

DEVELOPMENT OF FUZZY LOGIC AND NEURAL NETWORK CONTROL AND ADVANCED EMISSIONS MODELING FOR PARALLEL HYBRID VEHICLES

The National Renewable Energy Laboratory (NREL)
Golden, CO.

*Arun Rajagopalan
Gregory Washington
Giorgio Rizzoni
Yann Guezennec*

Center for Automotive Research
Intelligent Structures and Systems Laboratory

The Ohio-State University
Columbus OH 43210.
December 2001.



TABLE OF CONTENTS

TABLE OF CONTENTS	2
LIST OF FIGURES.....	3
LIST OF TABLES.....	4
ACKNOWLEDGEMENTS	5
EXECUTIVE SUMMARY	6
INTRODUCTION	8
INTELLIGENT CONTROL OF A HEV USING EMISSIONS CONSTRAINTS	9
INTRODUCTION	9
INTELLIGENT CONTROL – THE BIG PICTURE	9
<i>Instantaneous control</i>	10
<i>Why Fuzzy Logic?</i>	10
THE CONTROL STRATEGY	10
<i>Optimal Torque Computation</i>	12
<i>Effect of the Driver’s torque request and battery SOC</i>	18
PREDICTIVE CONTROL.....	22
<i>Future State Prediction Algorithm</i>	23
RESULTS	26
<i>Urban driving cycle</i>	26
<i>Highway cycle</i>	28
<i>US06 aggressive driving cycle</i>	29
WILLANS’S LINE MODEL OF AN INTERNAL COMBUSTION ENGINE.....	32
INTRODUCTION	32
MODELING.....	32
<i>Symbols Used:</i>	32
USAGE	34
RESULTS	35
<i>Example 1:</i>	35
<i>Example 2:</i>	36
IMPLEMENTATION IN ADVISOR 3.2.....	39
EMISSIONS CONTROL STRATEGY	39
PREDICTIVE CONTROL STRATEGY WITH EMISSIONS	40
<i>Sample Driving cycles</i>	41
WILLAN’S LINE MODEL.....	41
CONVENTIONS FOR VARIABLES	42
SUMMARY OF CONCLUSIONS	43
REFERENCES.....	45
APPENDIX.....	46

LIST OF FIGURES

Figure 1: Structure of the Powertrain Control System.....	10
Figure 2: Base block diagram with fuzzy controller	11
Figure 3: Fuzzy Controller block components.....	11
Figure 4: IC Engine emissions – a general trend	12
Figure 5: Finding an optimal IC Engine operating point	13
Figure 6: Optimal IC Engine torques based on individual weighting.....	15
Figure 7: Optimal IC Engine torque based on equal weighting.....	16
Figure 8: Sample cost function minimization at 1000 rpm.....	17
Figure 9: Sample cost function minimization at 3000 rpm.....	17
Figure 10: Optimal torque computation block diagram.....	18
Figure 11: Main fuzzy logic controller	19
Figure 12: Concept of operating about optimum torque	20
Figure 13: Visualization of the Fuzzy Controller action.....	21
Figure 14: Predictive control algorithm (supervisory control structure).....	22
Figure 15: Predictive controller inputs and output.....	23
Figure 16: Look-ahead window	23
Figure 17: Look-behind window.....	24
Figure 18: Emissions results - UDDS	27
Figure 19: Emissions results - HWFET	29
Figure 20: Emissions results – US06	30
Figure 21: Input – output relationship in an IC Engine.....	32
Figure 22: ICE fuel flow map for a scaled 3.0 L SI Engine.....	35
Figure 23: ICE Efficiency map for a scaled 3.0 L SI Engine.....	36
Figure 24: ICE fuel flow map for a scaled 4.0 L CI Engine	37
Figure 25: ICE efficiency map for a scaled 4.0 L CI Engine.....	38
Figure 26: Use of the Willan’s line model	41
Figure 27: Urban Dynamometer Driving Schedule (UDDS).....	46
Figure 28: Highway Fuel Economy Test (HWFET).....	47
Figure 29: US06 aggressive driving cycle	47

LIST OF TABLES

Table 1: Default weights in the control strategy	13
Table 2: Weights during a cold start – prioritize emissions.....	14
Table 3: Weights during high torque demand – compromise emissions	14
Table 4: Overview of the control strategy.....	21
Table 5: Rule base for predictive FLC	25
Table 6: Results - UDDS.....	26
Table 7: Results – HWFET	28
Table 8: Results – US06.....	29
Table 9: Files added in ADVISOR 3.2 – Emissions control.....	40
Table 10: Files added in ADVISOR 3.2 – Predictive control	41
Table 11: Sample driving cycles for the predictive algorithm	41
Table 12: Conventions for Variables	42

ACKNOWLEDGEMENTS

This work was funded by the Center for Transportation Technologies and Systems (CTTS) of the National Renewable Energy Laboratory (NREL), Golden, CO. We wish to thank Kenneth J. Kelly (Senior Engineer and point of contact, Systems Analysis Team, NREL) and Keith Wipke (NREL's Vehicle Systems Analysis Task Leader) for their continued support. Dr. Giorgio Rizzoni (Director, Center for Automotive Research and Intelligent Transportation, the Ohio-State University) and Dr. Yann Guezennec (Faculty, OSU) provided project oversight and technical expertise. Xi Wei offered a helping hand in the development of the Willan's Line model. We also thank students of the Center for Automotive Research, The Ohio-State University, for their support and ideas in the project.

EXECUTIVE SUMMARY

This report describes the development of new control strategies and models for Hybrid Electric Vehicles (HEV) by the Ohio-State University, on a contract with the National Renewable Energy Laboratory (NREL), Golden, CO. The contract was assigned to the Center for Automotive Research and Intelligent Transportation (CAR-IT) by the Center for Transportation Technologies and Systems (CTTS) at the National Renewable Energy Laboratory. The report indicates results from models created in NREL's ADVanced Vehicle SimulatOR (ADVISOR 3.2). Also presented are the results of a scalable IC Engine model, called the Willan's Line technique, implemented in ADVISOR 3.2.

The project was aimed at the development of Advanced Control Techniques for Hybrid Vehicles. Techniques such as Fuzzy Logic were used in the implementation of the complex control problem. An optimal torque split between the Internal Combustion Engine (ICE) and the Electric Motor is the outcome of the new algorithm. The new control algorithm also takes into account emissions from the IC Engine in calculating the torque produced. The Willan's line model was used to produce a scaled fuel flow map of an Internal Combustion Engine of any arbitrary size.

The following are the highlights of the new control algorithm:

1. Optimization structure for contending parameters such as emissions and fuel efficiency.
2. Flexibility in assigning a relative importance between the various emissions and fuel economy.
3. Non-Linear correction for IC Engine torque output based on battery State of Charge (SOC).
4. The scalable control strategy embedded in ADVISOR 3.2 can be used for all Parallel Hybrid Electric Vehicle configurations.

This document reveals that effective control of emissions can be performed even under demanding road conditions. An optimality issue exists when dealing with various emissions, but any specific emission characteristic can be effectively targeted by modifying the relative importance of that emission parameter.

Changing relative importance between efficiency and emissions during the vehicle operation is useful in curtailing excess output of a particular parameter at certain instances. An example would be to heavily penalize operating points with high emissions when the engine is still cold. This mechanism is found to effectively reduce cold start emissions.

Future work may include bringing other vehicle parameters into the optimization criteria such as battery State of Charge (SOC), etc. and the addition of intelligent algorithms that modify an existing control strategy based on information about the future route of the vehicle. Vehicle navigation systems these days can pinpoint the current location of the vehicle and allow planning routes to a destination. Information along that planned route may be used in the control strategy to aim at optimizing performance for an entire trip. These algorithms are currently being studied by faculty and students at the Ohio-State University.

An algorithm to produce the fuel-use maps of IC Engines is implemented in ADVISOR 3.2. This algorithm, called the Willan's Line Model, is based on the scaling of parameters from engines with similar characteristics. A database of the parameters of the Willan's Line model for SI and CI engines are included in the software. There is a provision for adding parameters of newer engines into the database if available. The advantage with this algorithm lies in the fact that the user obtains a fuel-use map, from entering some basic engine parameters such as stroke length, bore diameter, etc.

The programs associated with all the algorithms are integrated into ADVISOR 3.2, consistent with the format for all data files for ADVISOR.

INTRODUCTION

Hybrid Electric Vehicles (HEVs) are projected as one of the solutions to the world's need for cleaner and more fuel-efficient vehicles. The efficacy of an HEV lies in its control strategy. The diligent use of the 2 power sources, namely the internal combustion engine (ICE) and the electric motor (EM) determines the fuel consumption, the emissions output, and the charge-sustaining behavior of the vehicle, while maintaining drivability.

The work presented below, was performed with the support of the National Renewable Energy Laboratory (NREL), Golden, CO. The task is to develop and validate an Intelligent Control Strategy for a Parallel Hybrid Electric Vehicle using Emissions constraints in ADVISOR. It must be noted that the strategy is programmed for ADVISOR 3.2, in the form suitable for a backward simulator. A backward simulator is one in which the road-load experienced by a vehicle is translated through the drivetrain as a load on the power sources, namely the IC Engine and the electric motor. Implementing this strategy in an actual vehicle would require modification in the way the inputs are fed into the algorithm. But the core algorithm would still be based on the same concept. In actual implementation, there are other factors which affect the performance of the vehicle in terms of fuel economy and emissions outputs. The results of this algorithm are based on steady – state (static) data maps available in ADVISOR 3.2.

INTELLIGENT CONTROL OF A HEV USING EMISSIONS CONSTRAINTS

Introduction

An Internal Combustion Engine (IC Engine) produces harmful emissions as a by-product of the combustion process inside a cylinder. These emissions have been controlled effectively in a conventional vehicle by the diligent use of fuel-injection systems. But in a conventional vehicle, the torque and speed produced by the IC engine is dependent on the driver's request (effectively the road load). This prohibits the IC Engine from operating at its optimal operating point (optimal torque and speed) based on fuel economy and emissions, and is always at the driver's discretion. This is the main advantage of a Hybrid Electric Vehicle (HEV) where the IC Engine can be commanded to run near its optimal operating point, while load-balancing using a buffer, such as an electric motor. The following section describes an intelligent control strategy using Fuzzy Logic that helps reduce emissions and improve fuel economy.

Intelligent Control – the Big Picture

Numerous methods have been designed for choosing the operating point of an IC Engine in a parallel HEV. The concept presented in this section is based on choosing an operating point about an optimal operating point, based on the road load, the battery State of Charge (SOC), and the optimal operating point of an IC Engine. An optimal operating point (ideal case) is calculated based on the minimization of a criterion, of which fuel and emissions are contending parameters. Based on a set of weights, the relative importance of fuel economy and emissions is dynamically chosen, and an optimal torque that can be requested from the IC Engine is calculated. Based on the battery SOC constraints and the road load (driver's request), the actual output torque of the IC Engine is computed. The remaining torque (at that speed) required to meet the road load is provided by a buffer, the Electric Motor (EM). The buffer may produce either positive or negative torque. This is the well-known concept of load-leveling. The basic idea is schematically shown in Figure 1. The strategy is implemented in ADVISOR, a backward simulator. Thus, actual implementation will be different from the way control is performed. But the basic concepts of control will remain the same.

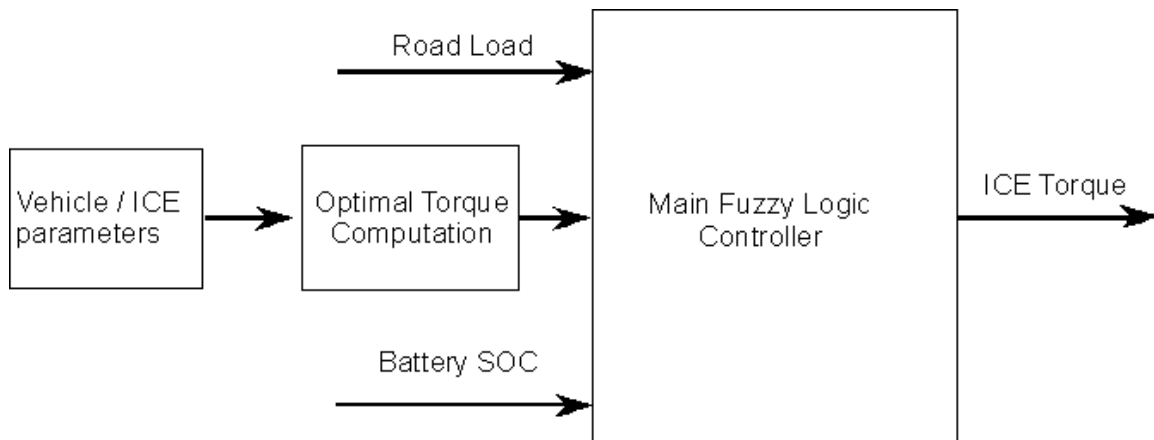


Figure 1: Structure of the Powertrain Control System

Instantaneous control

This control strategy is an instantaneous control strategy. It is based on IC Engine parameters at the current instantaneous vehicle speed. Most vehicles today are built around a powertrain containing a fixed-ratio transmission. (Manual or Automatic). The IC Engine nearly always rotates at a speed determined by the rotational speed of the wheels (except while the clutch is slipping). If a continuously variable transmission is available, then the speed of the IC Engine is also a degree of freedom in control. In this project, we deal with fixed-ratio transmissions, where IC Engine speed control is not performed. This control strategy can also be coupled with a speed control scheme (shift-control) if developed for a vehicle coupled to an automatic transmission.

Why Fuzzy Logic?

Intelligent control is performed with the help of Fuzzy Logic as a tool. Fuzzy Logic enables the development of rule-based behavior. The knowledge of an expert can be coded in the form a rule-base, and used in decision making. The main advantage of Fuzzy Logic is that it can be tuned and adapted if necessary, thus enhancing the degree of freedom of control. It is also a non-linear structure, and this is especially useful in a complex system such as an advanced powertrain. For more details on Fuzzy Logic Control (FLC), refer to Fuzzy Control by Passino and Yurkovich [7]. In essence a FLC is a natural extension of many of the rule based controllers implemented (via look-up tables) in many vehicles today.

The Control Strategy

The control strategy involves calculating the torque produced by the IC Engine based on various parameters such as road-load and battery SOC. This includes the calculation of an optimal torque based on contending IC Engine parameters, and deciding the actual torque output by later modifying the optimal torque based on road load and battery SOC. The control strategy is made scalable, and can be used with any parallel HEV model in

ADVISOR 3.2. The controller programs are created using MATLAB script files, and can be modified by the user. The following Figure 2 indicates the position of the Fuzzy Logic Controller in the Simulink block in ADVISOR.

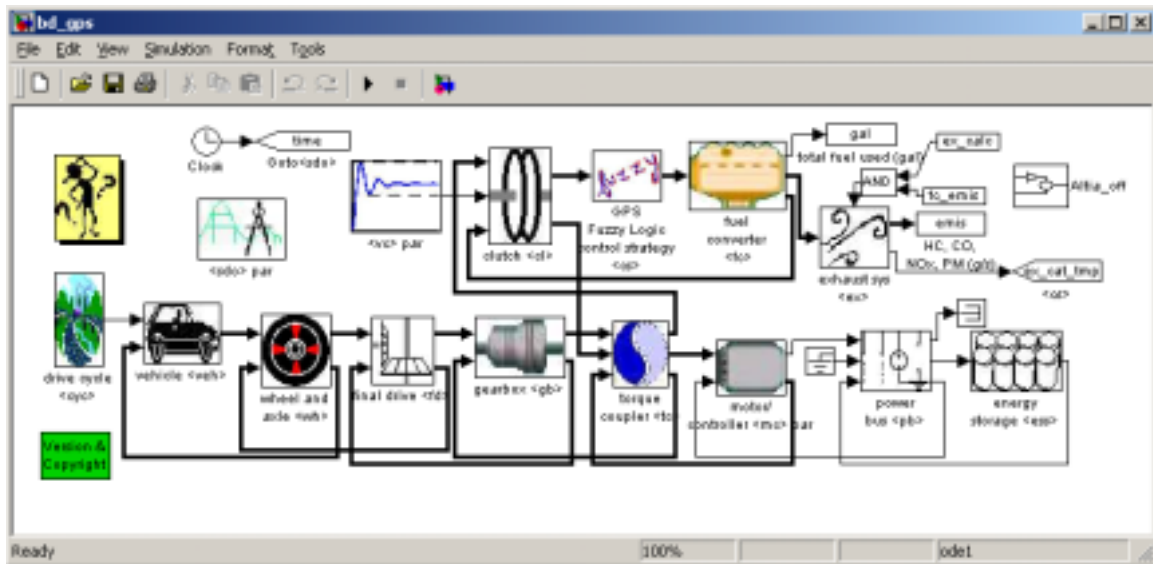


Figure 2: Base block diagram with fuzzy controller

The structure of the Fuzzy Controller is shown below in Figure 3.

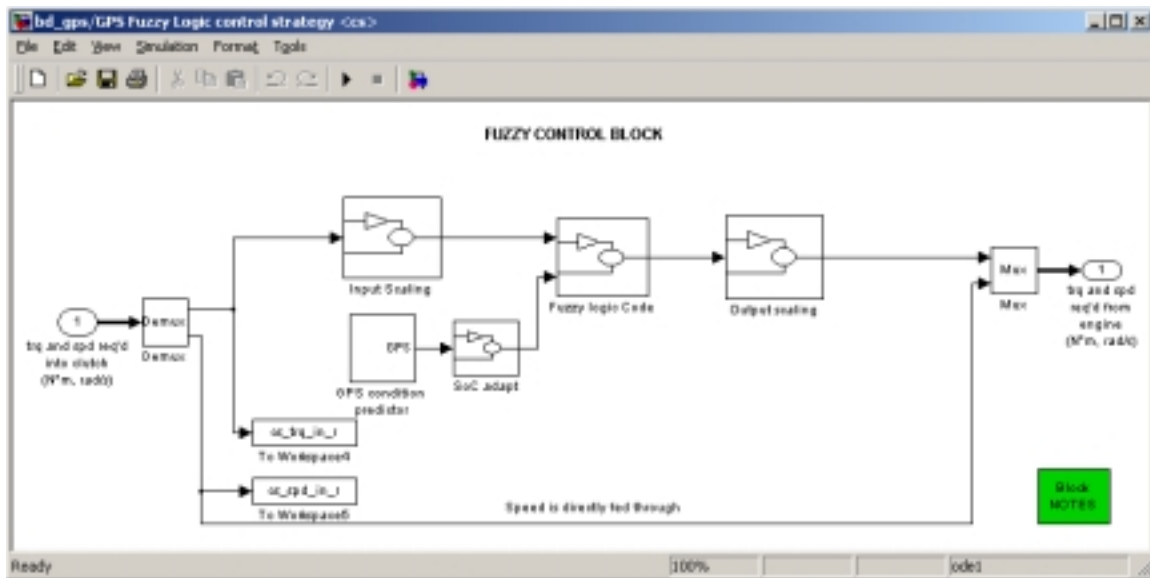


Figure 3: Fuzzy Controller block components

Optimal Torque Computation

The data for an IC Engine in ADVISOR 3.2 is in the form of a 2-dimensional map, indexed by torque and speed. Information regarding fuel economy (g/s) and emissions such as CO, HC and NO_x (g/s) is available for various speeds and torques. Shown below as a sample are the fuel use and emissions data for a 1.9L Turbo Diesel Engine available in ADVISOR.

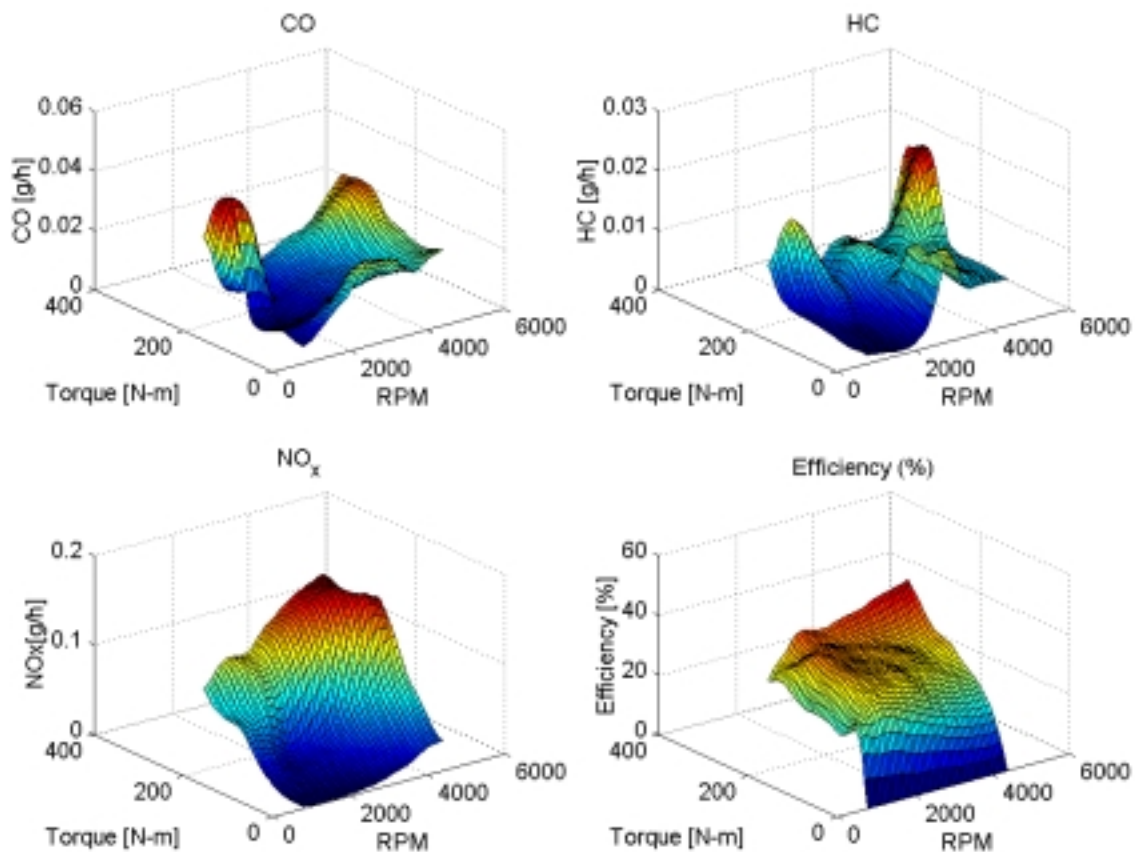


Figure 4: IC Engine emissions – a general trend

As seen in the above figure, we can determine the parameters at all torques for any given speed, up to the maximum torque point.

The process of finding an optimal point is shown in Figure 5 below.

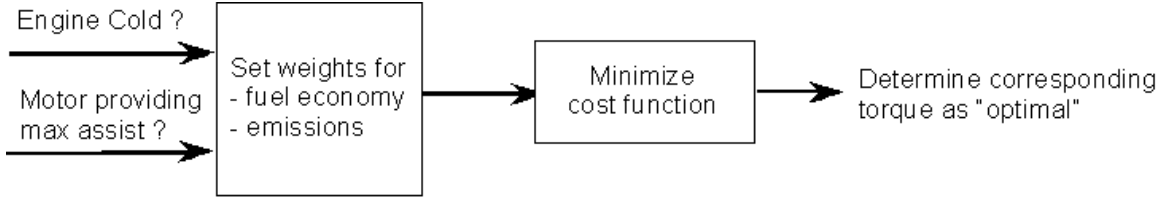


Figure 5: Finding an optimal IC Engine operating point

The four competing parameters in the determination of an optimum are fuel efficiency, NO_x , CO and HC emissions. At any particular point in time, the simulation determines a speed of rotation for the IC Engine (based on the powertrain configuration and the current gear ratio). This is the speed at which the “instantaneous” optimization is performed. For the current speed, all possible torques that the IC Engine can provide are considered. The four competing parameters for all torques at the current speed are taken from the data maps.

$$\min J = (w_1(1 - \overline{\eta}) + w_2 \overline{\text{NO}_x} + w_3 \overline{\text{CO}} + w_4 \overline{\text{HC}}) \quad (1)$$

where J = cost function
 w_i = weights, $i=1,2,3,4$
 $\overline{\eta}$ = Normalized efficiency
 $\overline{\text{NO}_x}$ = Normalized NO_x emissions
 $\overline{\text{CO}}$ = Normalized CO emissions
 $\overline{\text{HC}}$ = Normalized HC emissions

Here, the values are normalized with respect to the maximum for that particular speed. Relative weights are assigned to each parameter based on their importance. This is one large degree of freedom, and the weights must be selected for each IC engine based on their individual data maps. For our example in the report, we use the following default weights.

Parameter	Weight
Efficiency	0.7
NO_x	0.3
CO	0.1
HC	0.1

Table 1: Default weights in the control strategy

The above weights provide an optimum based heavily on efficiency and NO_x , while CO and HC are also considered. These weights can be varied during the operation of the vehicle, based on certain vehicle parameters.

COLD START: When the coolant temperature is still below acceptable values (cold start), emissions are very high, since the catalysts are not up to their operating temperatures yet. In this case, higher weights can be assigned to emissions. This increases the cost function near high-emission zones, and thus would be avoided when a minimum is found. This is seen to improve cold-start emissions considerably. The following are sample weights for a cold-start scenario.

Parameter	Weight
Efficiency	0.25
NO _x	0.25
CO	0.25
HC	0.25

Table 2: Weights during a cold start – prioritize emissions

INADEQUATE TORQUE: The control strategy is primarily used to control the IC Engine torque output. The electric motor is used to perform load-leveling, i.e. provide extra torque or absorb excess torque, according to the driver's torque demand. In case we determine that the motor has been providing its maximum torque during the previous time step, it most probably indicates that the IC Engine and the motor assisting together have not met the driver's torque request. This warrants some increase in IC Engine torque, so as to satisfy the driver's torque demand. In this case, efficiency is given higher priority, since high efficiencies occur near higher torque regions. It must be noted that this case may not work for all IC Engines. For engines whose peak efficiency at each speed is not near high torque, these weights cannot be used.

Parameter	Weight
Efficiency	1
NO _x	0.1
CO	0.1
HC	0.1

Table 3: Weights during high torque demand – compromise emissions

These are sample weights and can be modified to suit a particular IC Engine. The following plots show the effect of choosing the weights on the position of the optimal torque point.

The following plots in Figure 6 show the optimum operating torques for the contending parameters taking them individually into consideration. This is obvious when we refer to Figure 4, which shows us a 3-D map of the same parameters. The dotted lines indicate the optimal torques at the various speeds based on the weights given therein.

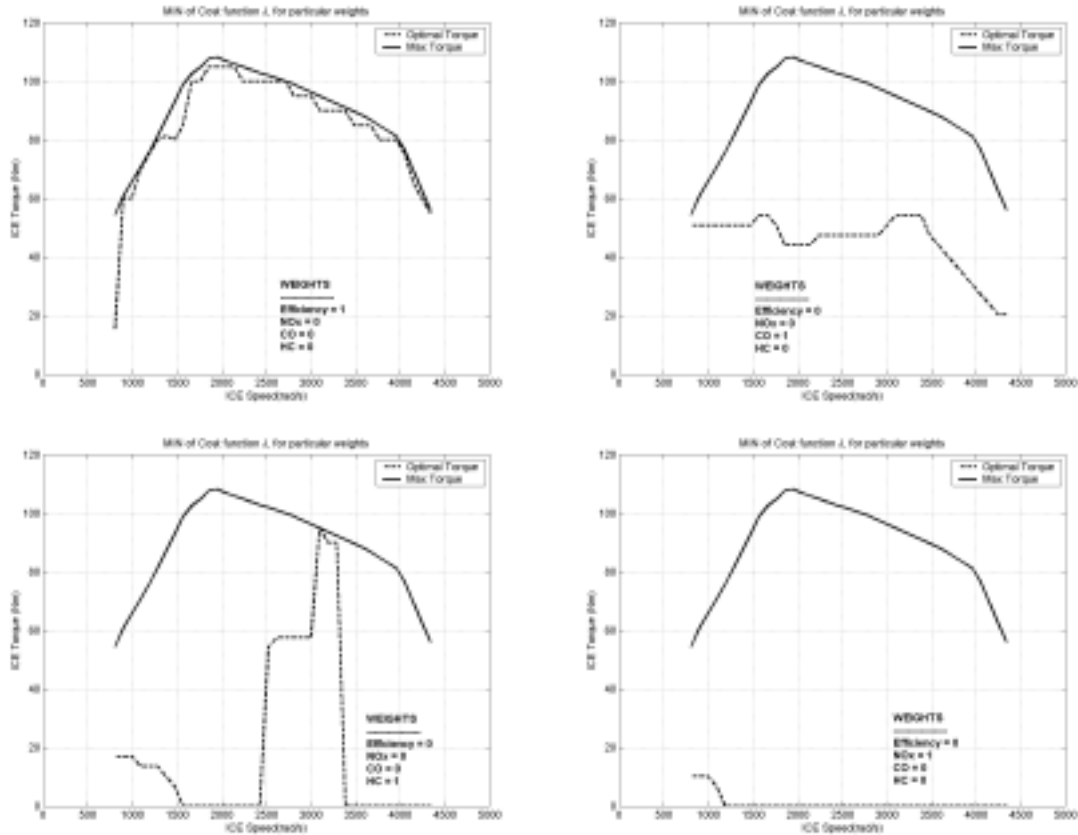


Figure 6: Optimal IC Engine torques based on individual weighting

Some inferences can be made from taking the above plots to indicate a general trend:

- We cannot operate always at the peak efficiency point, since it is bound to lie near the maximum torque envelope (Wide open throttle line). This would lead to excessive fuel consumption and heavy charging due to load-leveling. It may result in damage to the battery, if excess charging is not controlled.
- We cannot optimize for NO_x alone, since NO_x appears to be lowest at zero torque values.

Note: An optimal solution is always a compromise. Since fuel economy and emissions are usually contending parameters, there is a trade-off obtained between these parameters. But from simulations, we can obtain a fairly considerable decrease in emissions with only a negligible loss in fuel economy. Another important advantage of this method is the ability to control any one of the parameters decisively, by increasing their relative weights.

If we equally weight the importance of the four contending parameters, then we get an optimal curve as given in Figure 7.

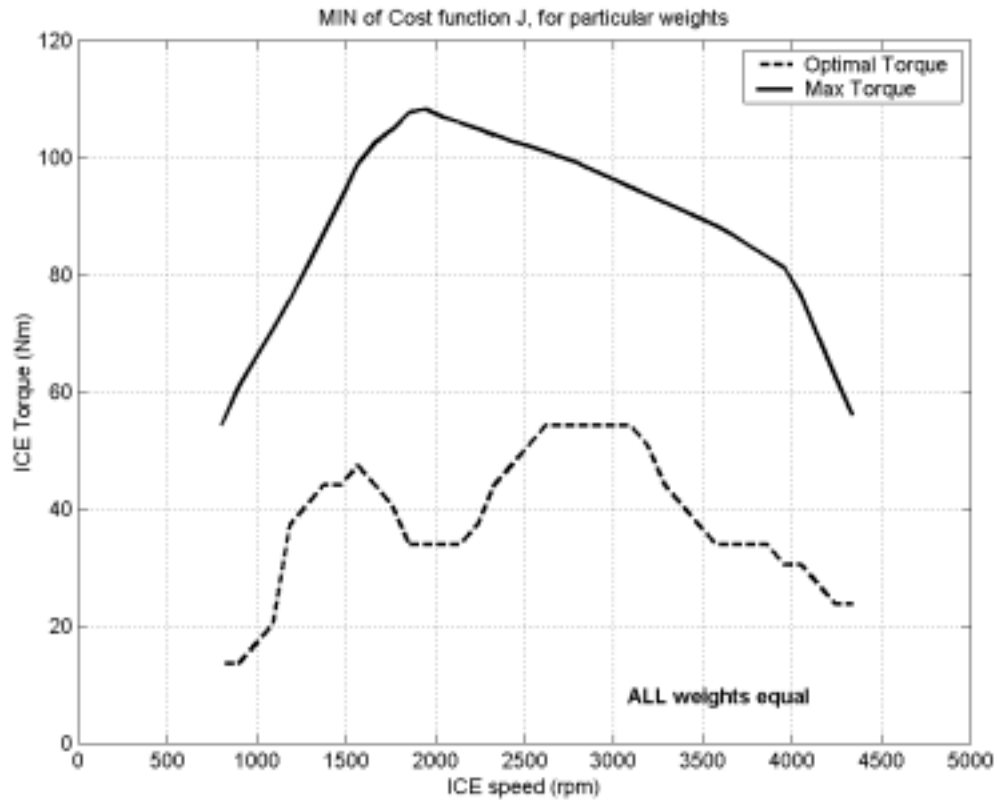


Figure 7: Optimal IC Engine torque based on equal weighting

When a cost function is calculated for a particular speed, a minimum is found for all torques from zero to the maximum torque at that speed. The following figures show us the cost function and the minimum for a particular set of weights given in Table 1.

Figure 8 shows us the cost function for the Optimal point at 1000 rpm. Figure 9 below shows the cost function and minimum (optimal) at 3000 rpm.

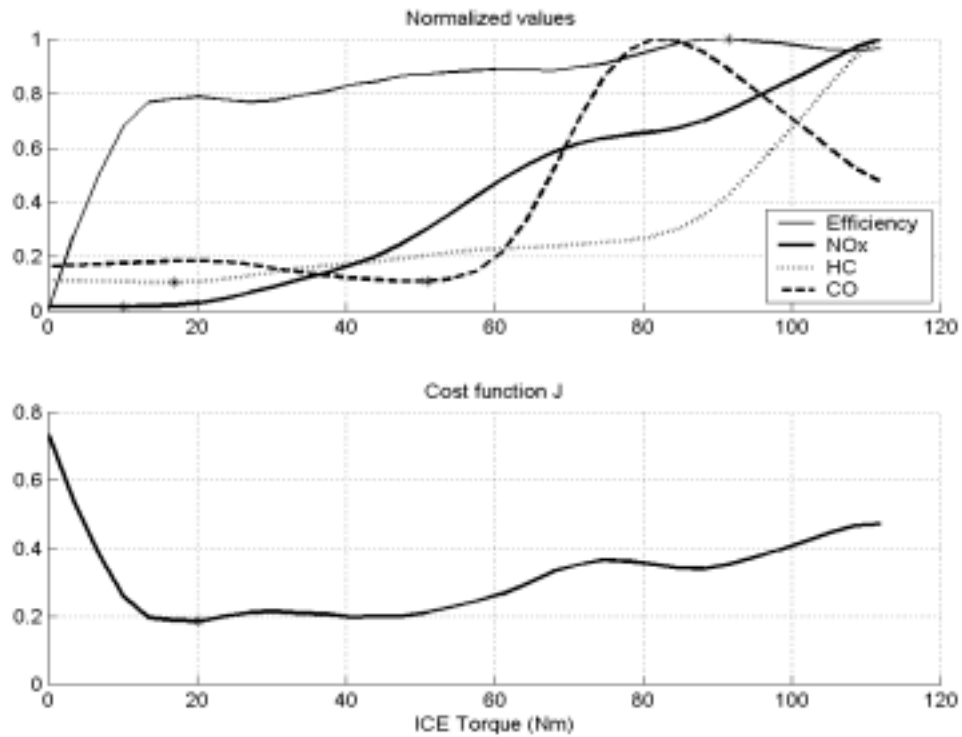


Figure 8: Sample cost function minimization at 1000 rpm

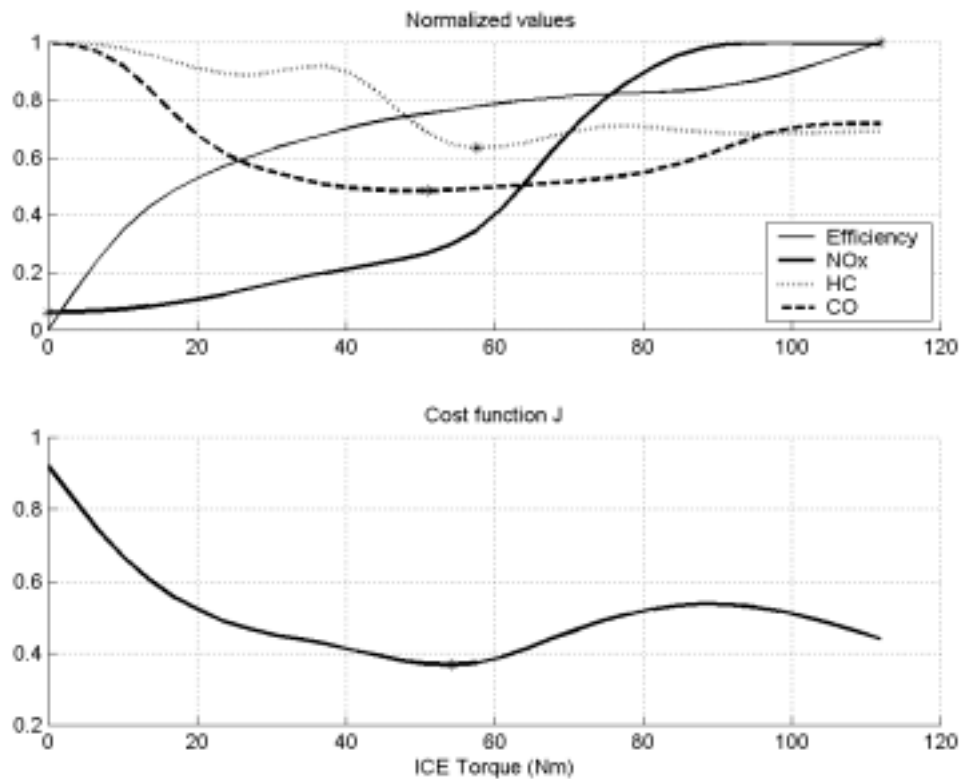


Figure 9: Sample cost function minimization at 3000 rpm

From the above plots, we see that an optimal torque based on our definition is calculated. It must be noted that IC Engine data in ADVISOR may be given for points above the maximum torque envelope. This algorithm takes the optimal, considering only the region between zero and maximum torques.

The Simulink block associated with the optimal torque calculation is shown in Figure 10.

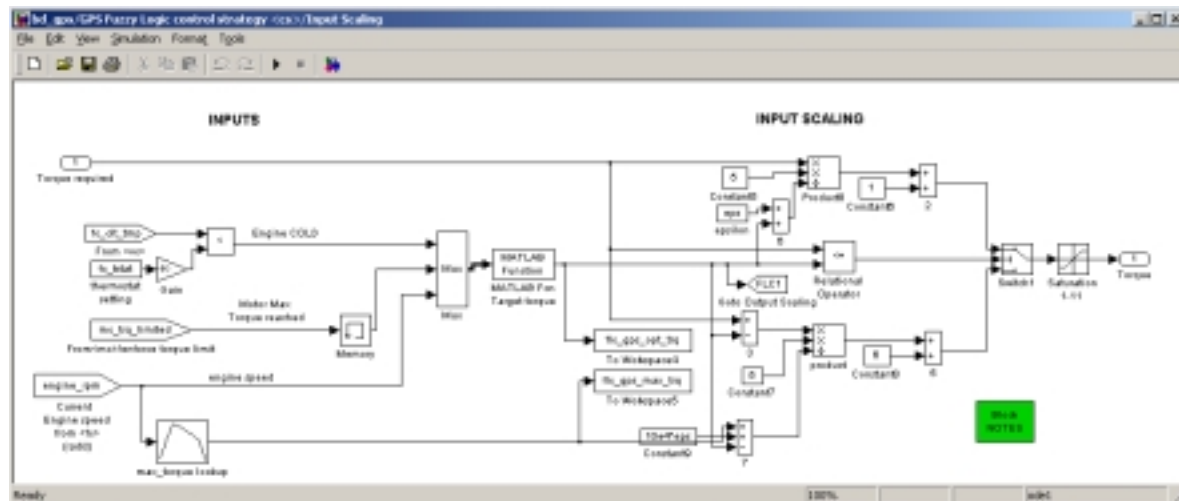


Figure 10: Optimal torque computation block diagram

This optimum is fed in to the Fuzzy controller, where the driver's demand and the battery SoC are also considered. Running the engine at the calculated optimal torque may not satisfy either the driver's demand or the charge-sustaining requirement of the HEV. Thus, adjustment of this 'optimal' torque is necessary, and is discussed in the following section.

Effect of the Driver's torque request and battery SOC

In powertrain control, the bottom line is that the vehicle must follow the driver's command. Torque requested from the powertrain must be met at all times. If the driver demands very low torque (indicated by small throttle pedal angles), such as while cruising, and the optimum IC Engine torque at that speed is near high torques, the IC Engine must be commanded to operate below optimum, to satisfy the road load and also to prevent the Electric Motor from generating large negative torques. This also depends on the battery SOC. In the above case, the higher the battery SOC, the less capable the Electric Motor is of generating electricity and thus load-leveling the IC Engine. That is where the main Fuzzy Logic controller is actively used.

Similarly, if the driver demands very high torque, such as during an acceleration, and the optimum found based on the weights at that speed happens to lie near low torques, deviation from the optimal is warranted. This is due to the fact that satisfying the driver's demand is the primary concern. In this case, if the battery is sufficiently charged, the IC Engine can still be allowed to operate at lower torques (near optimal) and the Electric Machine can be used to assist. If the battery SOC does not look promising, then deviation

from the optimal point by a large amount is inevitable. These non-linearities and exceptions are handled by the fuzzy logic controller.

A schematic of the main FLC is shown in Figure 11.

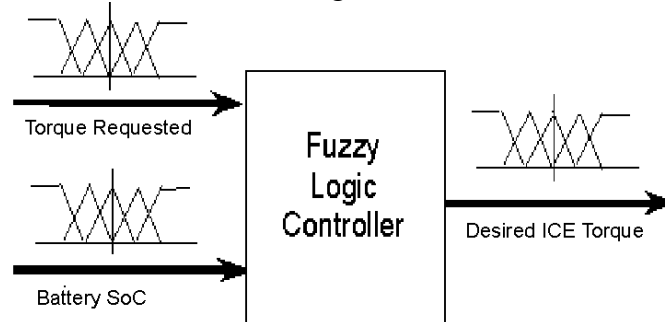


Figure 11: Main fuzzy logic controller

Fuzzy Logic Control for SoC Correction

A Fuzzy Logic Controller (FLC) is used to decide the operating point of the ICE. It follows the idea of *load leveling*, where the EM is used to provide assist or generate, while running the ICE at an optimum. The idea behind this approach is to vary an optimal ICE torque, based on SOC constraints. This concept has been explored in other ways, such as by Baumann et al [1]. One may refer to [1] for an in-depth explanation of fuzzy logic control and its application to HEV control. Care is taken in the rule-base to make sure the FLC does not shift too much from this optimal point so as to compromise the overall efficiency of the system. It must be remembered that one of the constraints is to remain charge-sustaining, and this is implemented in the form of the user inputting certain SOC limits within which operation is desired.

Figure 12 depicts the operating principles of the main FLC. The FLC receives 2 inputs, the required driveline torque (engine + motor) and the battery pack SOC. Each input and output has 11 triangular membership functions, giving a total of 121 rules. The desired driveline torque is first fed as a request to the IC Engine, and is scaled from '1' to '11', '1' representing zero torque, '6' representing the calculated 'optimal' ICE torque and '11' representing maximum ICE torque at that speed. To elaborate further, a driveline torque input of '1' indicates that the driver request torque that is very low, near zero torque output of the IC Engine. If the driver's demand is at the calculated 'optimal' ICE torque, the FLC input value is set to '6'. If the driver's demand is '11', it indicates that the required driveline torque is at the maximum IC Engine torque (at that speed), or higher (due to saturation of fuzzy membership functions at 11). To perform the scaling of the driveline torque (first input to the FLC), it is compared to the calculated 'optimal' torque at each time step. It must be noted that the calculated 'optimal' torque (value '6') itself varies every time step, based on the weights and the minimization criterion. Similarly, SoC is scaled from '1' to '11', '1' corresponding to the lowest SOC limit and '11' corresponding to the highest SOC limit, as set by the user. The general logic behind the main FLC is described in Table 4. Of course, the actual simulated controller is in more detail, with 121 rules, and displays some non-linear behavior.

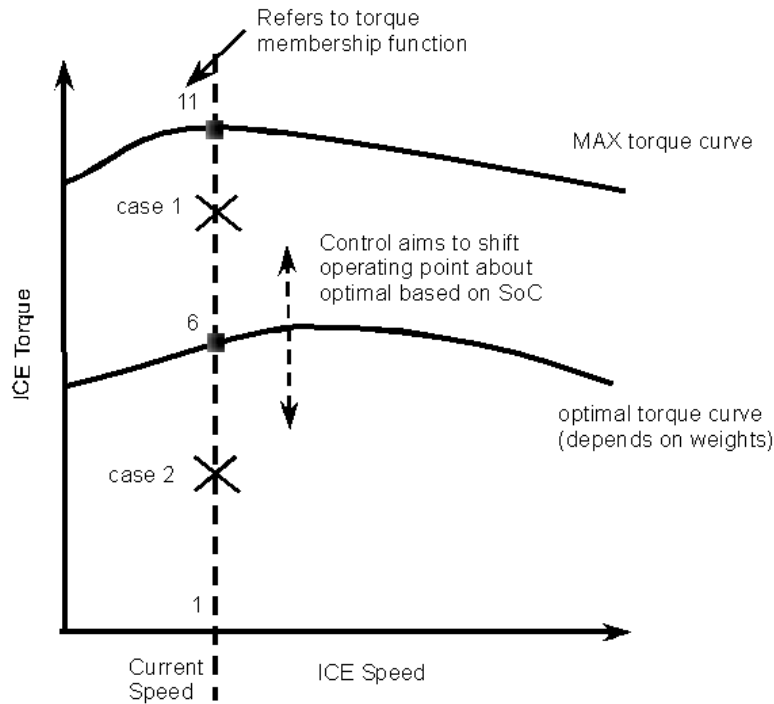


Figure 12: Concept of operating about optimum torque

In Fuzzy Logic terminology, the *universe of discourse* defines the range of values that an input or an output can take. In our case, the *universe of discourse* can take values from ‘1’ to ‘11’. In our control strategy, we modify the scale of our *universe of discourse* of the first input to the FLC (driveline torque), based on the calculated ‘optimal’ torque. Further reference on Fuzzy Logic and the associated terminologies can be found in Fuzzy Control by Passino and Yurkovich [7].

Example case 1 – Let us consider a case, when the required torque is above the optimal torque point, as seen by the “X” in Figure 12 (case 1). Let us assume that the SOC is high. We would like to bring the ICE operating point near the optimal torque point marked ‘6’ (for that speed). This would mean a lower torque output by the ICE than what is required to meet the driver’s demand. This requires that the EM be run as a motor to make up for the remaining torque, provided there is enough battery charge. Since we do have sufficient charge in this case, the ICE is allowed to operate near the optimal torque point.

Example case 2 – Let us consider another case, when the desired torque is below optimum at point 2 (‘X’ in Figure 12), and the battery SoC is very low. We would like to increase the ICE torque output, and bring it near the optimum. This would require load-leveling by the EM (EM functioning as a generator), so as to output only what the driver demands. This is possible only if the SOC is not high. In our case, we have low SOC, and thus we can run the EM as a generator, while running the ICE at the optimum. In both the cases, the EM torque is represented by $T_{EM} = T_{Total} - T_{ICE}$. In this case, T_{EM} is negative, generating some energy into the battery pack.

Inputs	Low SoC	Optimum SoC	High SoC
Desired torque lower than optimal	$\uparrow T_{ICE}$ EM as gen.	$\uparrow T_{ICE}$ near optimum EM as gen (small)	$\downarrow\downarrow T_{ICE}$ EM as motor
Desired torque is optimal	$\uparrow T_{ICE}$ EM as gen	$\leftrightarrow T_{ICE}$ \leftrightarrow EM	$\downarrow T_{ICE}$ EM as motor
Desired torque higher than optimal	$\uparrow\uparrow T_{ICE}$ EM as gen	$\downarrow T_{ICE}$ near optimum EM as motor (small)	$\downarrow T_{ICE}$ EM as motor

Table 4: Overview of the control strategy

The structure in Table 4 is used for this FLC. The “*desired ICE torque* “ can be summarized as a SoC adjusted optimal ICE torque, based on efficiency and emissions along a particular speed.

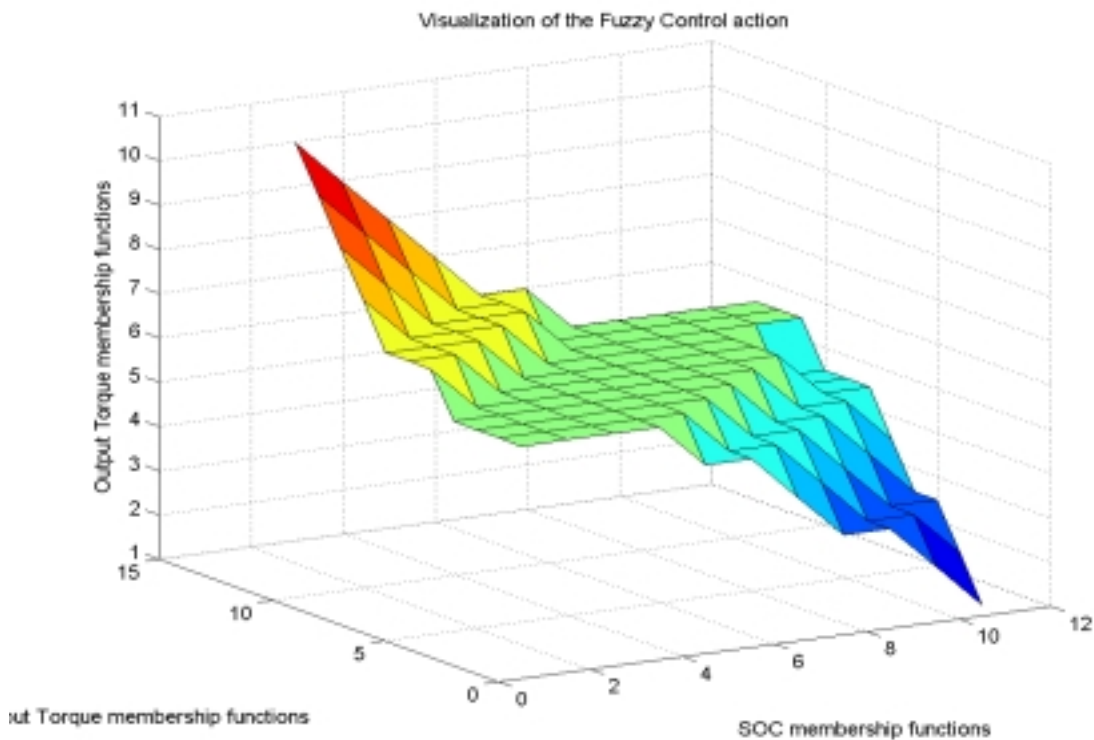


Figure 13: Visualization of the Fuzzy Controller action

Predictive control

Ideally, it is possible to minimize fuel and emissions along a trip by finding a global optimum if we know the entire trip information before hand. But in reality, we have no such facility. This problem can be circumvented with knowledge of the type of obstacles that will be faced in the near future, such as heavy traffic, or a steep grade etc. We can then perform control actions now, to account for those situations in the future. For example, if we are on a highway entering a city where we expect to encounter heavy traffic, it is useful to charge the batteries now to be able to use the motor later for city driving (possibly drive all-electric). Global Positioning Systems (GPS) have become a commonplace in vehicle navigation systems, installed in many vehicles around the world. Most systems pinpoint the location of the vehicle overlaid on top of a map of the surrounding area, along with traffic information, and topographical data such as elevation etc. Newer systems like General Motor's OnStar navigation system [8] allow the user to plan a route to a destination and also obtain information along that path. We propose an algorithm to implement HEV control making use of the wealth of data available about the surroundings and specifically along a planned route.

A predictive control strategy is being developed at Ohio-State the aim of achieving a higher degree of control over the emissions and fuel economy over an entire trip of a vehicle. Most Geographical Information Systems (GIS) have topographical information which contain details about the neighboring terrain. If a trip is planned on such a navigation system, information regarding the elevation along the route ahead can be accessed from a GIS database (perhaps available on a CD) and can be used in determining the change in elevation along that route. When coupled with speed predictions from traffic conditions along that route (highway / city traffic), this set of information can be used to perform control at the current instant, to prepare for adversities in the future. A general overview of the concept is shown in Figure 14.

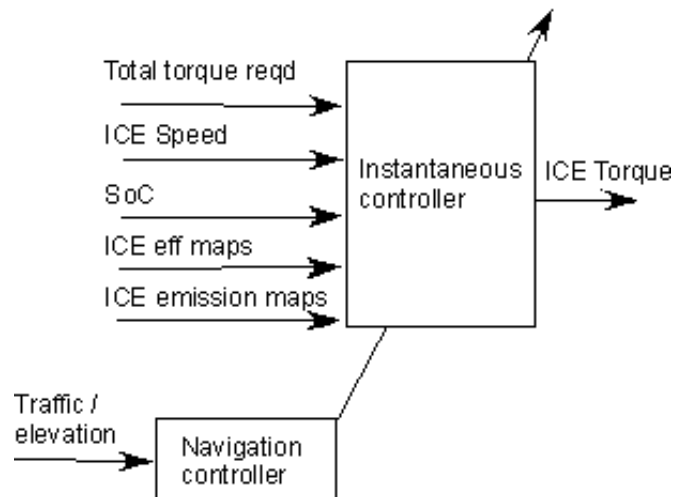


Figure 14: Predictive control algorithm (supervisory control structure)

Future State Prediction Algorithm

Figure 14 displays the supervisory control system structure, which is explained below. An instantaneous control strategy is used to determine the optimal ICE torque contribution for the current speed. This is based on the ICE efficiency and emission maps, the battery state of charge and the total driveline torque required, as described earlier in this report. Another controller called the navigation controller or predictive controller varies the instantaneous controller parameters based on implied future states. A Fuzzy logic controller is used in each of the 2 cases, to enable rule-based behavior and to implement non-linear control. The inputs to the navigation controller are speed and elevation of sampled points along a pre-determined route, from a navigation system. This controller decides to perform control action now, based on the recent history of the motion of the vehicle and the changes to this motion in the near future. An example of this would be to charge the batteries now, if a hill is to be encountered in a few minutes, so as to allow the vehicle to climb up at acceptable speeds then. This concept is currently being developed, and the files / programs for this module will be available to NREL soon. A symbolic representation of the predictive controller is shown in Figure 15.

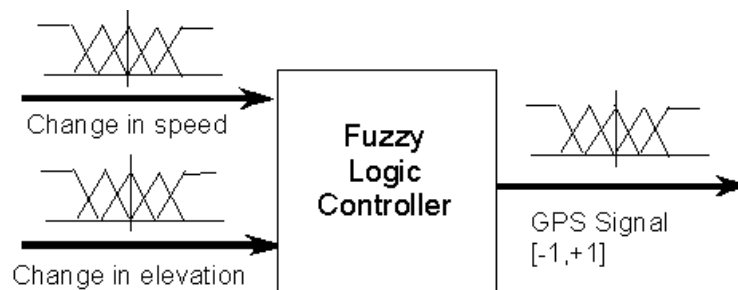


Figure 15: Predictive controller inputs and output

PREDICTION OF STATE – A Fuzzy Logic Controller (FLC) is used to predict the future state of the vehicle. The information supplied regarding the future is a sampled set in a look-ahead window along a planned route. Assuming a planned route ahead of us, we determine a look-ahead window, along which we sample to obtain elevation and traffic information. Traffic information is in the form of speed (of traffic flow) at those sampled points.

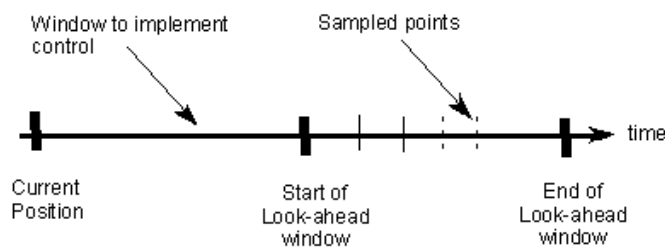


Figure 16: Look-ahead window

Figure 16 shows the region of look-ahead for the prediction algorithm. The value of look-ahead length and the number of samples in the region are set based on optimal values for a given drive cycle and perhaps vehicle configuration.

SAMPLED POINT – At each sampled point, we can obtain traffic information (speed of vehicles) and elevation at that point. The speed state of the vehicle in the look-ahead zone is taken as the average of the sampled points. Similarly, the elevation of the look-ahead zone is taken as the average of the elevation at the sampled points.

$$Future_State = mean(sampled_states) \quad (2)$$

PREDICT CHANGE – To implement control, we need to determine if the future vehicle states vary from the current state. The predictive algorithm cannot be implemented in all cases. For example, there may be no gain in charging the batteries for future city traffic, if the vehicle is already in heavy traffic. To determine the current vehicle state, a weighted average of the past velocities over a look-behind interval is calculated. In actual implementation, the previous n-speeds can be stored in memory and a weighted average calculated. The look-behind time scale is easy to determine, since a time averaged value is used, and most recent velocities will have a higher impact.

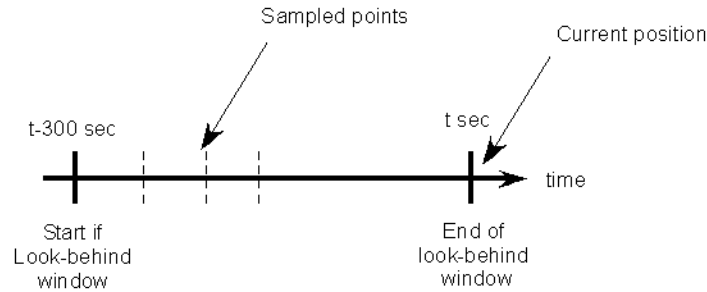


Figure 17: Look-behind window

Note that in the first few seconds of a trip, we assume that there is not enough information to determine the vehicle speed, and thus the prediction algorithm is switched off initially.

$$Recent_Speed = \frac{\sum weight_i * velocity_i}{\sum weight_i} \quad (3)$$

The inputs to the fuzzy logic controller are the change in vehicle speed corresponding to recent speeds and the change in elevation. The change in elevation is calculated with respect to the current vehicle elevation.

Note that in implementation, if at any point, even if traffic information service may not be available, the prediction algorithm based on elevation alone can be implemented since navigation systems can carry terrain data from a Geographic Information System (GIS).

Figure 15 shows the FLC with 2 inputs and a “GPS signal” as the output. This “GPS signal” is designed to carry a range from +1 to –1. The aim of the prediction algorithm is to tell the main controller to charge or discharge the batteries now, to account for future vehicle states. This “GPS signal” is used to indicate the control logic, with ‘+1’ indicating a desire to charge the batteries now, for future heavy EM use (such as stop-and-go traffic). Similarly, a ‘–1’ indicates a desire to deplete the batteries now, in the hope that the EM will receive regenerative energy soon along the planned route (such as a downhill route).

NAVIGATION FLC CONTRUCTION– The FLC used in the above algorithm has 2 inputs and 1 output. Each input and output has 11 triangular membership functions, giving a total of 121 rules. For the change in speed, ‘1’ corresponds to slower traffic ahead, and ‘11’ corresponds to faster traffic ahead. Similarly, for the change in elevation, ‘1’ corresponds to a steep down grade, while ‘11’ corresponds to a steep climb. The minimum condition is used for the premise calculation and Center of Gravity (CoG) is used for the output inference. The range of the universe of discourse for each of the 2 inputs and 1 output is scalable. This is useful in tuning the fuzzy controller to different requirements. The rules base is created based on heuristics on how a vehicle should react to oncoming states. The following is a general overview of the rule base.

Condition AHEAD	Going downhill	Constant elevation	Going uphill
Slower traffic (city)	Do nothing	Charge	Charge more
Similar traffic	Discharge	Do nothing	Charge
Faster traffic (highway)	Discharge more	Discharge	Do nothing

Table 5: Rule base for predictive FLC

Table 5 shows the general trend of the rules in the controller. Detailed non-linear behavior is implemented in the actual rule base, which is an 11x11 matrix.

A sample rule - For example, consider the case when the navigation system indicates a “Downhill grade”, and “heavy (slower) traffic” ahead. It is more efficient to use the EM in slower traffic, and thus a higher batter charge in the future is desired. This may warrant running the EM as a generator now. But a downhill grade into the ‘future zone’ will automatically create an opportunity to regenerate some energy. Thus, our controller does not command any specific charge now from the EM.

UPDATE FREQUENCY – The predictive algorithm from the navigation system is used to vary the torque split control strategy, but is implemented only at discrete times. The algorithm updates must occur at optimal times only. An algorithm that updates very

frequently might defeat the purpose of prediction, not giving enough time for control. This time is a variable in simulation, and we may choose to update the control every look-ahead period. In implementation, the distance along the planned route can be taken as a factor in dynamically choosing this look-ahead window.

Currently (as of December 2001), this algorithm is still being tuned, and thus will be made available as soon as it is completed. For further details, please refer to Rajagopalan et al [10].

Results

To demonstrate our control algorithm considering efficiency and emissions, a sample small parallel HEV with the following parameters is chosen. All the data are taken from existing sources in ADVISOR 3.2.

- 34 kW CI Engine (scaled from 1.9L 67 kW Volkswagen ICE tested by ORNL)
- 25 kW AC Solectria ACgtx20/AC300 Induction Electric Motor
- Hawker Genesis 26 Ah Lead Acid Battery, 25 modules
- Vehicle Mass = 1266 kg
- Coeff. of drag = 0.335
- Frontal area = 2 m²

The various parameters for the control strategy are entered as variables and are implemented in Simulink blocks. Static maps for efficiency and emissions of the ICE are used.

We compare the emissions and fuel economy for various cycles. Some of the results are shown in the following sections.

Urban driving cycle

When tested with the urban driving cycle (UDDS), we see a significant improvement in the NO_x emissions as well as some improvement in fuel economy. The emissions figures are compared in Table 6.

Control Strategy	Fuel Economy (mpg)	CO Emissions (g/mi)	HC Emissions (g/mi)	NO _x Emissions (g/mi)
ADVISOR's Parallel Electric Assist strategy	59.3	0.113	0.044	0.988
New fuzzy logic control strategy	63.9	0.107	0.036	0.784

Table 6: Results - UDDS

A visual comparison of emissions is shown in Figure 18.

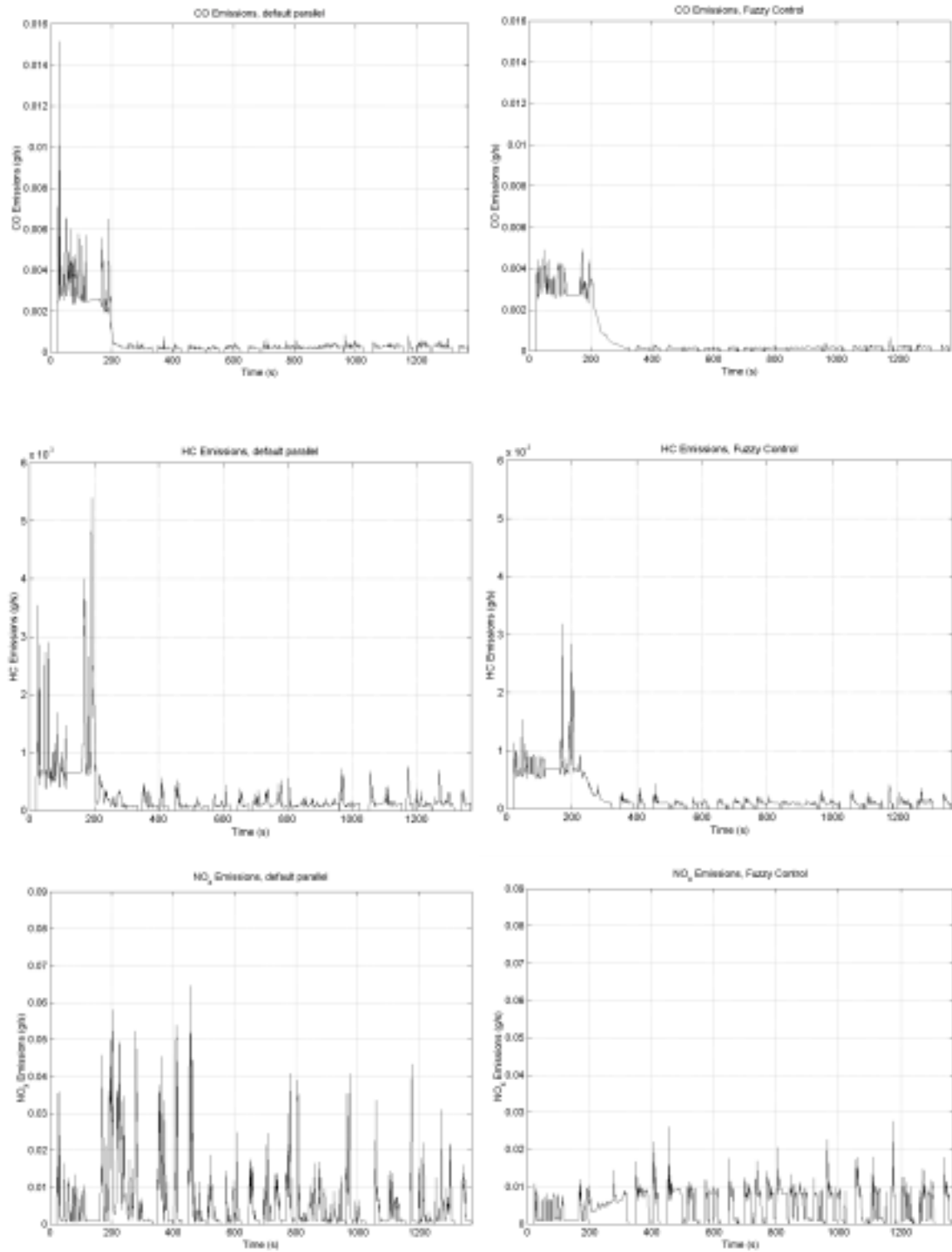


Figure 18: Emissions results - UDDS

It is observed that many of the high emissions regions seen in the plots on the left side (default parallel strategy) are avoided in the plots on the right side (fuzzy control). The reduction in cold-start emissions is also achieved.

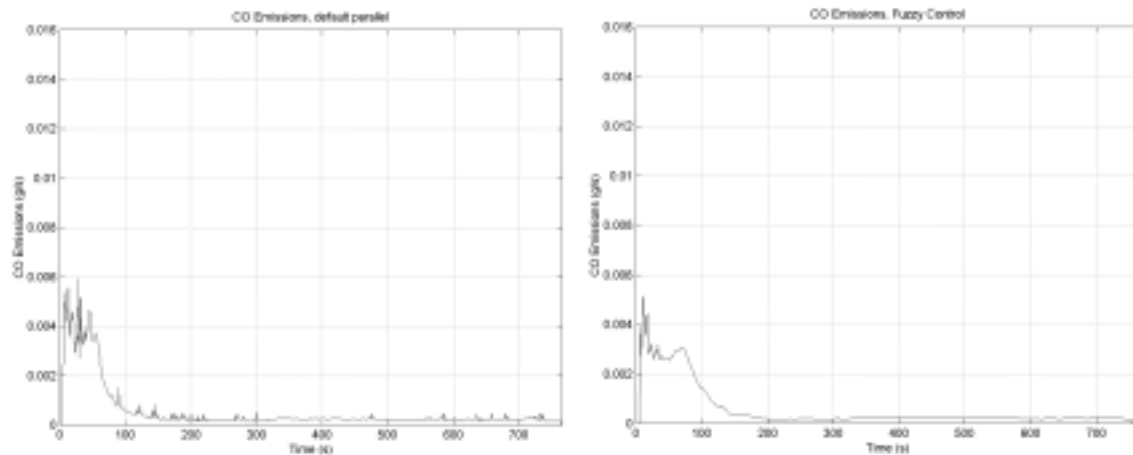
Highway cycle

In the highway cycle also, we see a significant improvement in the NO_x emissions, with about the same fuel economy.

Control Strategy	Fuel Economy (mpg)	CO Emissions (g/mi)	HC Emissions (g/mi)	NO _x Emissions (g/mi)
ADVISOR's Parallel Electric Assist strategy	68.2	0.043	0.023	0.82
New fuzzy logic control strategy	70	0.042	0.018	0.523

Table 7: Results – HWFET

The following plots indicate the emissions from the highway cycle, for the base control strategy and the Fuzzy Logic control strategy.



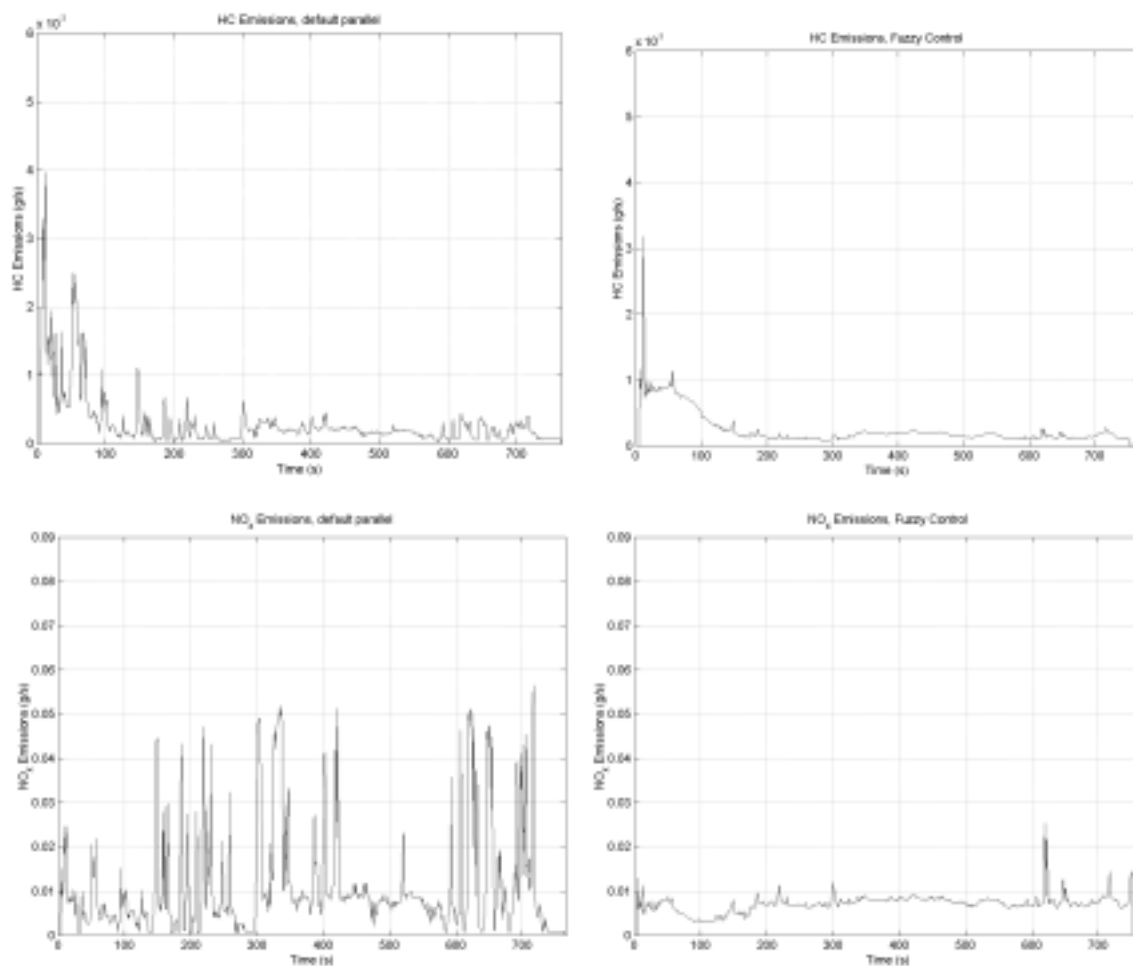


Figure 19: Emissions results - HWFET

As seen above in Figure 19, we see a large improvement in the NO_x emissions. Cold-start emissions are reduced, as expected.

US06 aggressive driving cycle

The new algorithm is compared with the base strategy when run over the US06 driving cycle, and the results are tabulated in Table 8.

Control Strategy	Fuel Economy (mpg)	CO Emissions (g/mi)	HC Emissions (g/mi)	NO _x Emissions (g/mi)
ADVISOR's Parallel Electric Assist strategy	48	0.045	0.033	2.179
New fuzzy logic control strategy	48.7	0.047	0.034	1.639

Table 8: Results – US06

The resulting emissions are plotted below.

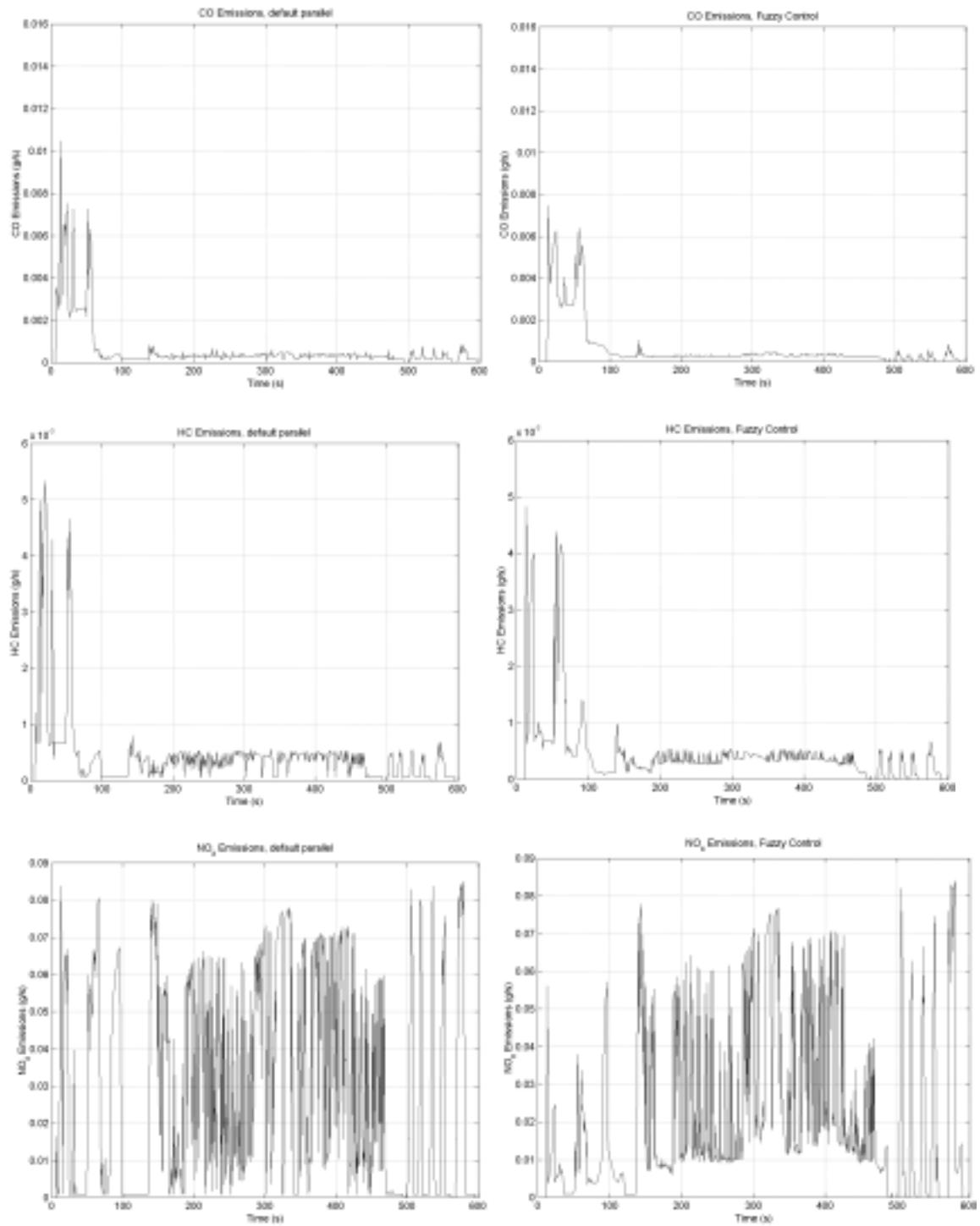


Figure 20: Emissions results – US06

As seen in Figure 20 and Table 8, NO_x emissions can be reduced without much compromise on fuel economy.

The change in overall SOC was considered in the above results. As expected, both strategies were able to charge sustain between the sample range of 0.55 to 0.45 SOC. The order of change in SOC for the 2 strategies were similar. It is observed from these results that it is easy to control NO_x emissions, as long as we avoid very high torque operating points, and points near the wide-open-throttle (WOT) range. But when it becomes necessary to charge the batteries or to produce extra torque, NO_x emissions have to be compromised. The ability to vary the weights in the strategy enables us to prioritize the emissions of our choice.

WILLANS'S LINE MODEL OF AN INTERNAL COMBUSTION ENGINE

Introduction

The Willan's line model is a scaling technique used to create numerical models of IC Engines of virtually any desired size. It is a good *quasi-static* representation of the fuel efficiency of CI/SI Engines. Its advantages lie in its scalability within engines of a certain class. However, this technique does not take into account several aspects of IC engines such as detailed engine geometry, fuel injection, thermal effects, engine dynamics, etc. Also, emissions are not modeled using this technique. Details on the Willan's Line model can be looked up on Xi Wie et al [8].

Modeling

In the Willan's Line Model, the energy conversion efficiency is represented as the ratio between the input and output power. The schematic in Figure 21 shows the affine relationship between input and output power.

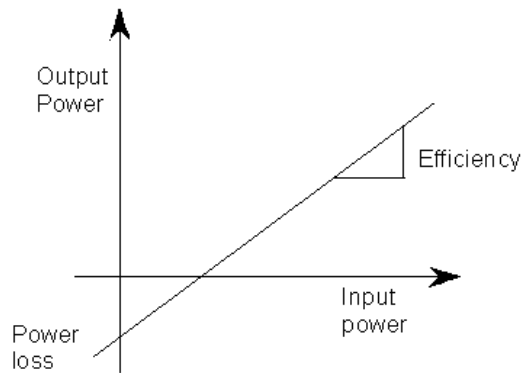


Figure 21: Input – output relationship in an IC Engine

The basic idea is to non-dimensionalize certain engine parameters such as torque and speed, so as to obtain a linear relationship between torque, speed and fuel use. The speed of the engine is represented as '*mean piston speed (m/s)*' instead of 'rpm' and torque is represented as '*mean effective pressure (N/m²)*' instead of 'Nm'. The basic empirical equations used in the Willan's line technique are as follows:

Symbols Used:

P_{ME} - Mean Effective Pressure

P_{MF} - Mean Fuel Pressure

P_{MR} - Mean Friction Pressure

C_M - Mean Piston Speed
 V_d - Engine displacement
 η_e - Engine Efficiency
 η_f - Fuel conversion Efficiency
 T_e - Torque produced by the ICE
 a_{ij} - Coefficients of fuel-conversion efficiency
 P_{MRi} - Coefficients of Friction pressure
 P_{MAXi} - Coefficients of Max. torque production
 Q_{LHV} - Lower calorific value of fuel
 \dot{m}_f - fuel flow rate (g/s) – used in ADVISOR data maps

Mean effective Pressure:

$$P_{ME} = \frac{4\pi T_e}{V_d} \quad (4)$$

Mean Piston Speed:

$$C_m = \frac{S\omega}{\pi} \quad (5)$$

Fuel Conversion Efficiency:

$$\eta_f = (a_{00} + a_{01}C_m + a_{02}C_m^2) - (a_{10} + a_{11}C_m)P_{MF} \quad (6)$$

Mean friction Pressure:

$$P_{MR} \cong P_{MR0} + P_{MR2}C_m^2 \quad (7)$$

Maximum Torque envelope:

$$P_{MAX} = P_{MAX0} + P_{MAX1}C_m + P_{MAX2}C_m^2 + P_{MAX3}C_m^3 \quad (8)$$

Mean Fuel Pressure:

$$P_{MF} = \frac{(P_{ME} - P_{MR})}{\eta_f} \quad (9)$$

Engine Efficiency (Used in the ADVISOR data files)

$$\eta_e = \frac{P_{ME}}{P_{MF}} \quad (10)$$

Fuel flow map (final result, used in ADVISOR)

$$\dot{m}_f = \frac{P_{MF} V_d \omega}{4\pi Q_{LHV}} \quad (11)$$

Using the above equations, we calculate the efficiency map of the IC Engine based on the availability of scaling coefficients. Scaling coefficients for a set of CI and SI Engines are obtained from researchers at the Center of Automotive Research (CAR) at the Ohio-State University. Scaling is best done when a given engine size is scaled from another engine of similar size, and more importantly, similar speed range, to avoid problems due to extrapolation. Data for more engines can be added to the database, since it is in the form of a matrix. Further, the database is split between SI and CI engines. To add more data from other engines to the database, one can use the Willan's Line algorithm to produce a set of coefficients for that particular engine, and add them to the database. The database is defined in the file FC_WILLANS_LINE.M.

Usage

To use the modeling feature, the user is required to input the following parameters pertaining to the desired IC Engine.

- Type of Engine (SI / CI)
- Cylinder bore diameter (m)
- Cylinder stroke length (m)
- Number of cylinders (--)
- Idle speed (rpm)
- Red Line (rpm)

The algorithm then calculates the volume of the engine. Based on the speed range and volume, the algorithm scales an existing engine from the database and produces a fuel-

flow map, which can be used in the appropriate Simulink block to calculate the fuel consumption of the IC Engine.

Results

Sample hypothetical IC Engine parameters are used in the program and the corresponding scaled data maps for fuel flow (in g/s) are produced. The results of these sample engines are given below:

Example 1:

Type: SI Engine
Capacity: 3.0 L
Idle: 700 rpm
Redline: 5000 rpm

The following fuel flow map is produced:

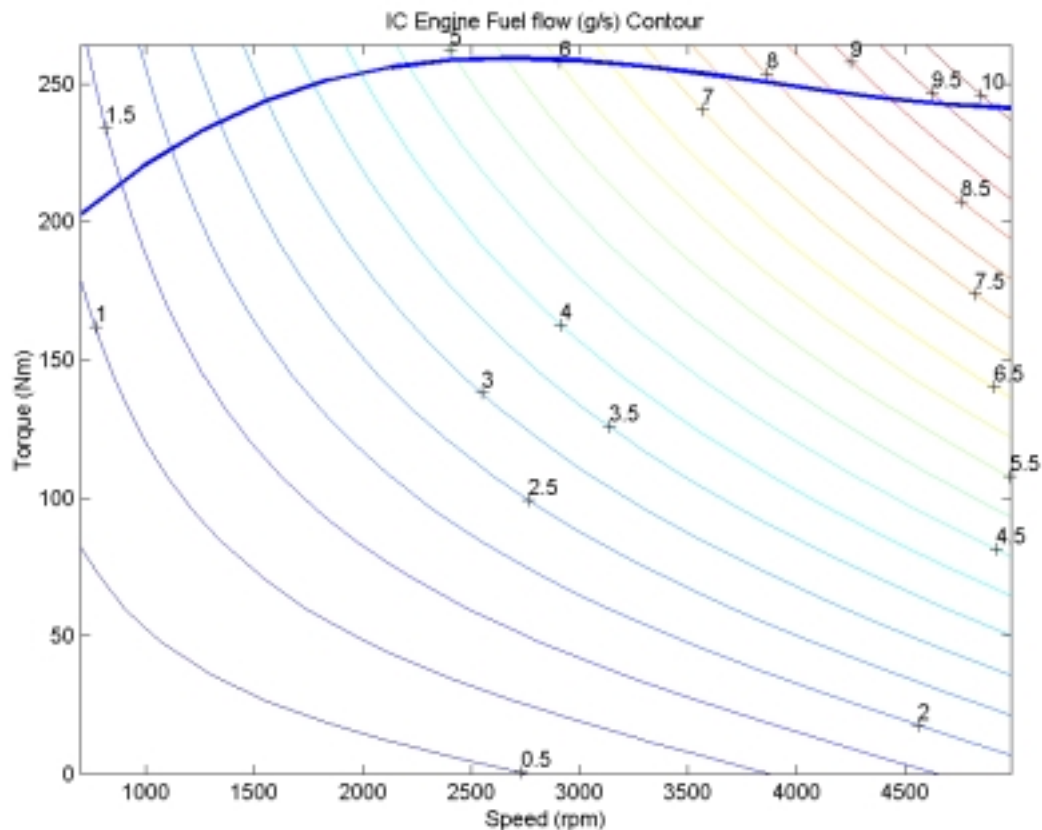


Figure 22: ICE fuel flow map for a scaled 3.0 L SI Engine

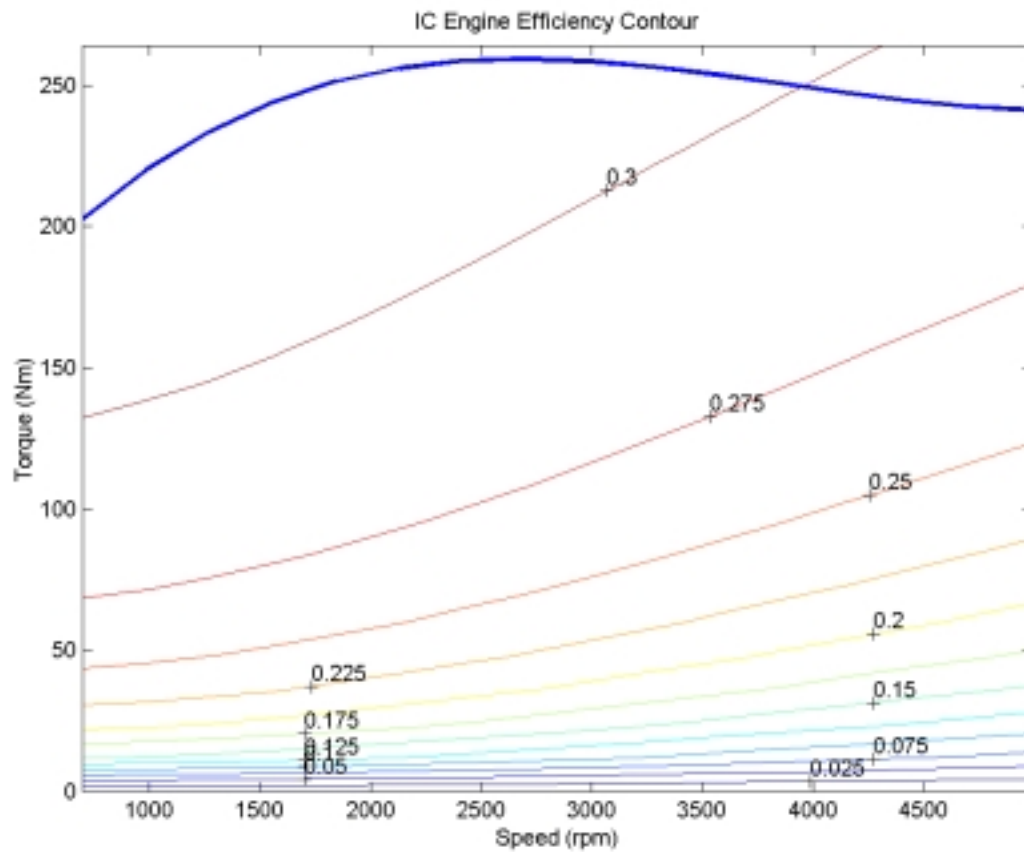


Figure 23: ICE Efficiency map for a scaled 3.0 L SI Engine

The fuel flow map is shown in Figure 22. It is seen that from simple IC engine parameters such as dimensions and engine type, we obtain a reasonable fuel-flow map, scaled from a database of Engines.

Example 2:

Type: CI Engine
Capacity: 4.0 L
Idle: 700 rpm
Redline: 4000 rpm

The following fuel flow map is produced:

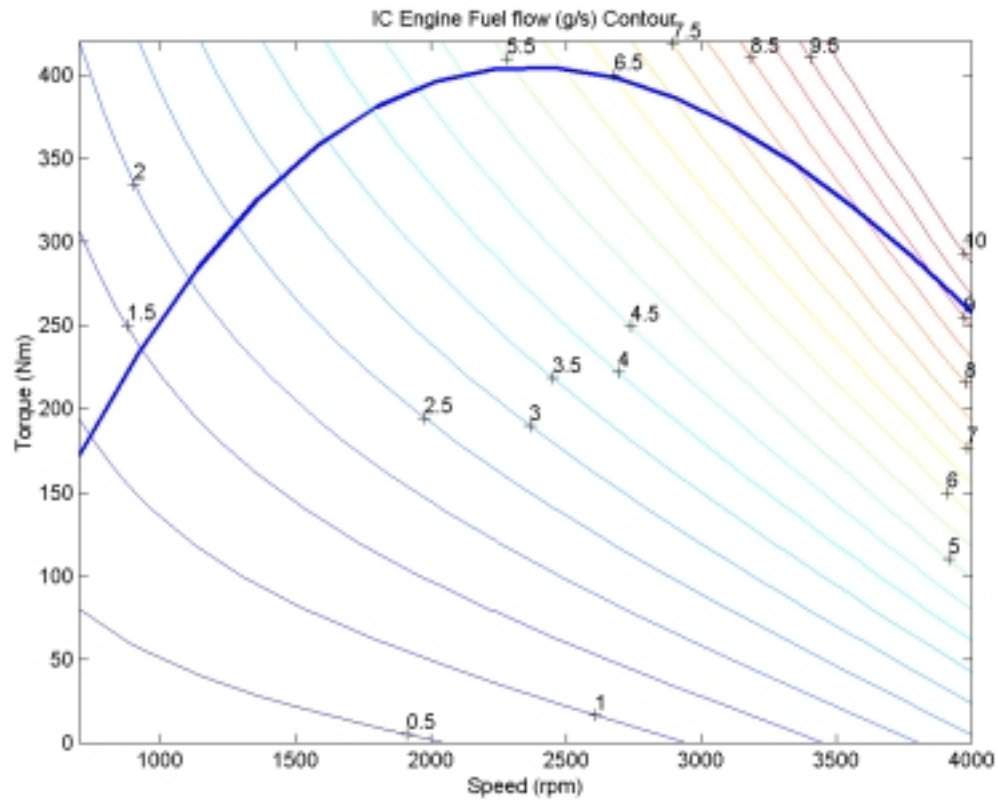


Figure 24: ICE fuel flow map for a scaled 4.0 L CI Engine

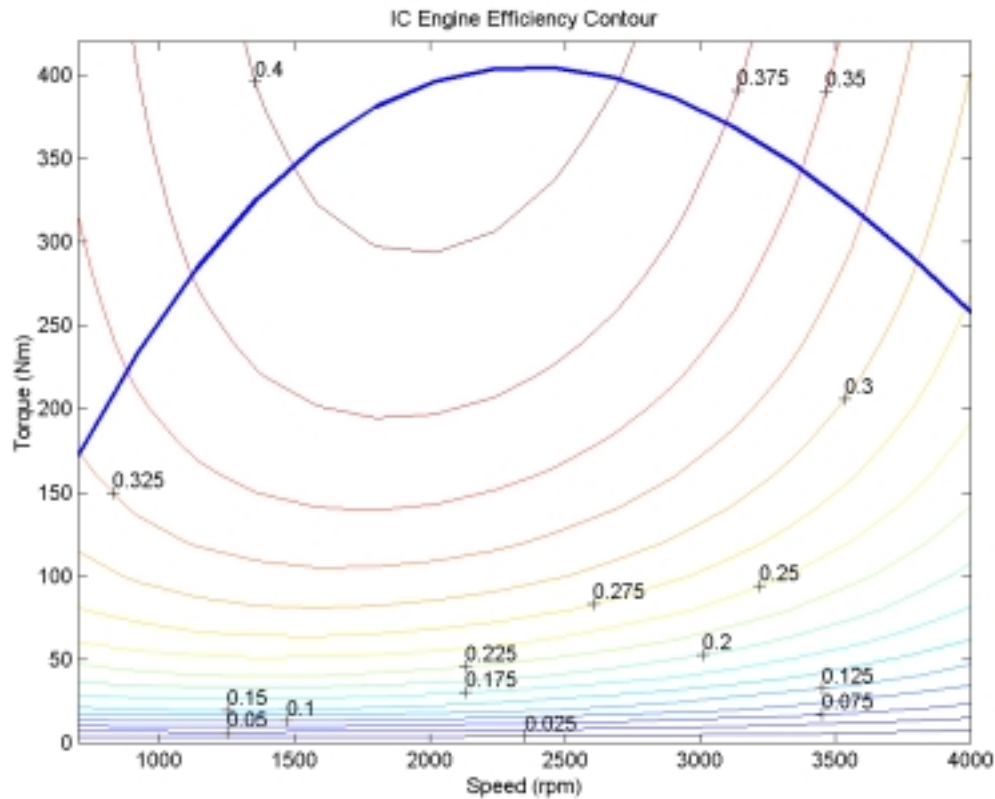


Figure 25: ICE efficiency map for a scaled 4.0 L CI Engine

The fuel-flow map (in g/s) for a 4.0 L CI Engine, scaled from similar engines is shown in Figure 24. As already seen, an acceptable fuel flow map is produced, with the specification of just a few IC engine parameters.

NOTE: The Willan's Line model is a technique used to produce the fuel-flow map for IC engines by scaling existing coefficients / parameters of IC Engines. It is not guaranteed that the fuel flow map and / or the efficiency map are accurate. Some IC Engine parameters obtain more accurate results than others. The model should be used diligently, and should be applied only when the scaled results are reasonable.

IMPLEMENTATION in ADVISOR 3.2

A total of 3 sets of changes are made to ADVISOR 3.2. First, there is a set of files (PTC_FUZZY_EMISSIONS.M control strategy and BD_FUZZY_EMISSIONS.MDL Simulink Block diagram) added for the Fuzzy Logic strategy with emissions (without Predictive control).

Also, development on a predictive control strategy based on GPS information is underway at the Ohio-State University. As a framework for this Predictive control strategy, a new set of files (PTC_FUZZY_EMISSIONS_GPS.M control strategy and BD_FUZZY_EMISSIONS_GPS.MDL Simulink Block diagram) is created, but the core predictive algorithm is not provided, and will be released when totally completed.

A new *fuel_converter* file is added to implement the Willan's line model for IC Engines. (FC_WILLANS_LINE.M)

To implement the algorithms in ADVISOR, some changes and additions are made to the data files. All files and variables are coded and named according to the conventions followed in ADVISOR 3.2. The conventions used are explained henceforth.

Emissions Control Strategy

A sample vehicle is defined with a choice of powertrain parameters. The Emissions control strategy contains the following files and variables.

File name	Function
fuzzy_emissions_in.m	Input file in ADVISOR 3.2 which contains the emissions control strategy vehicle configuration to be loaded with the GUI.
bd_fuzzy_emissions.mdl	The main block diagram that is used when the new fuzzy logic controller (ptc_fuzzy_emissions.m) is selected from the GUI interface
fuzzy_target_compute_emissions.m	Calculate the optimal torque for a given IC Engine speed, based on a set of weights. Also varies the weights based on vehicle parameters.
ptc_fuzzy_emissions.m	This file contains the powertrain control variables, including the battery SOC limits. When selected, uses the file bd_fuzzy_emissions.mdl to simulate the vehicle.
mfuzzy_emissions.m	The main Fuzzy Logic Controller which takes in the optimal torque and SOC as inputs and gives the actual torque as the output. This is used to shift the calculated optimal torque point based on the driver's request and the Battery SOC. Care is taken

	to control the deviation to a minimum, since a large correction from optimal destroys the purpose of optimization. The rule-base can be modified by the user if desired.
Block_diagram_name.m	Modifications are made to this file so as to enable the respective fuzzy logic Simulink blocks to be used, when the fuzzy powertrain controller is selected.

Table 9: Files added in ADVISOR 3.2 – Emissions control

Predictive Control Strategy with Emissions

For the predictive controller, the framework is supplied in ADVISOR 3.2. The main algorithm is under development, and the m-file with the algorithm will be released when completed. A sample vehicle is defined with a similar choice of powertrain parameters as the previous section (without predictive control). The predictive control strategy contains the following files and variables.

File name	Function
fuzzy_emissions_gps_in.m	Input file in ADVISOR 3.2 which contains the emissions control strategy vehicle configuration to be loaded with the GUI.
bd_fuzzy_emissions_gps.mdl	The main block diagram that is used when the new fuzzy logic controller (ptc_fuzzy_emissions_gps.m) is selected from the GUI interface
fuzzy_target_compute_emissions.m	Calculate the optimal torque for a given IC Engine speed, based on a set of weights. Also varies the weights based on vehicle parameters.
ptc_fuzzy_emissions_gps.m	This file contains the powertrain control variables, including the battery SOC limits. When selected, uses the file bd_fuzzy_emissions.mdl to simulate the vehicle.
mfuzzy_emissions.m	The main Fuzzy Logic Controller which takes in the optimal torque and SOC as inputs and gives the actual torque as the output. This is used to shift the calculated optimal torque point based on the driver's request and the Battery SOC. Care is taken to control the deviation to a minimum, since a large correction from optimal destroys the purpose of optimization.
Block_diagram_name.m	Modification are made to this file so as to enable the respective fuzzy logic Simulink blocks to be used, when the fuzzy powertrain controller is selected.
gps_predict.m	Currently a dummy file which outputs a null value.

	This file will contain the actual predictive control parameters for implementation when released.
--	---

Table 10: Files added in ADVISOR 3.2 – Predictive control

All the MATLAB and SIMULINK files created for the contract are designed in similar lines to existing .M files in ADVISOR 3.2. Comments and explanations are included where applicable.

Sample Driving cycles

To test the predictive strategy, 4 sample driving cycles are created from existing driving cycles, to simulate change in traffic or elevation for the route ahead. These 4 files are to be used only with the predictive control strategy. They are labeled as follows:

File name	Function
CYC_GPS1.M	Combines the HWFET, NREL2Vail, UDDS and the Vail2NREL cycles.
CYC_GPS2.M	HWFET followed by the UDDS
CYC_GPS3.M	UDDS followed by the HWFET
CYC_GPS4.M	UDDS with arbitrary elevation

Table 11: Sample driving cycles for the predictive algorithm

Willan's Line Model

For the Willan's line model, we have an M-file which contains a sample configuration, and the parameters for generating a new efficiency map. The user can use the template and create his own *Fuel_converter* input file.

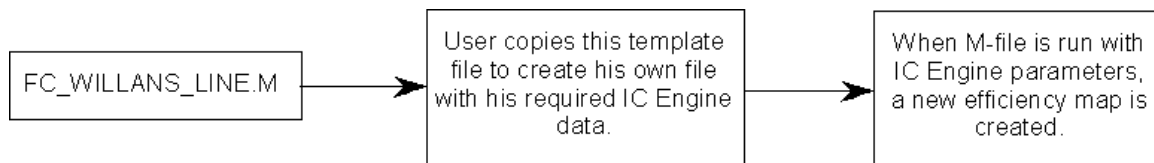


Figure 26: Use of the Willan's line model

Conventions for Variables

The conventions for naming variables used in the 3 new algorithms are described below in Table 12.

Variable Prefix	Function
flc_*	Used in the Fuzzy Logic Control algorithm
gps_*	Used in the predictive control algorithm. Many GPS_* variables are Global variables.
fc_wlm_*	Used in the Willan's Line model of an IC Engine
fc_wlm_* ci	Willans Line model data pertaining to CI engines
fc_wlm_* si	Willans Line model data pertaining to SI engines

Table 12: Conventions for Variables

SUMMARY OF CONCLUSIONS

This report describes the development of new control strategies and models for Hybrid Electric Vehicles (HEV) by the Ohio-State University, on a contract with the National Renewable Energy Laboratory (NREL), Golden, CO. The contract was assigned to the Center for Automotive Research and Intelligent Transportation (CAR-IT) by the Center for Transportation Technologies and Systems (CTTS) at the National Renewable Energy Laboratory. The report indicates results from models created in NREL's ADVanced Vehicle SimulatOR (ADVISOR 3.2). Also presented are the results of a scalable IC Engine model, called the Willan's Line technique, implemented in ADVISOR 3.2.

The project was aimed at the development of Advanced Control Techniques for Hybrid Vehicles. Techniques such as Fuzzy Logic were used in the implementation of the complex control problem. An optimal torque split between the Internal Combustion Engine (ICE) and the Electric Motor is the outcome of the new algorithm. The new control algorithm also takes into account emissions from the IC Engine in calculating the torque produced. The Willan's line model was used to produce a scaled fuel flow map of an Internal Combustion Engine of any arbitrary size.

The following are the highlights of the new control algorithm:

1. Optimization structure for contending parameters such as emissions and fuel efficiency.
2. Flexibility in assigning a relative importance between the various emissions and fuel economy.
3. Non-Linear correction for IC Engine torque output based on battery State of Charge (SOC).
4. The scalable control strategy embedded in ADVISOR 3.2 can be used for all Parallel Hybrid Electric Vehicle configurations.

This document reveals that effective control of emissions can be performed even under demanding road conditions. An optimality issue exists when dealing with various emissions, but any specific emission characteristic can be effectively targeted by modifying the relative importance of that emission parameter.

Changing relative importance between efficiency and emissions during the vehicle operation is useful in curtailing excess output of a particular parameter at certain instances. An example would be to heavily penalize operating points with high emissions when the engine is still cold. This mechanism is found to effectively reduce cold start emissions.

Future work may include bringing other vehicle parameters into the optimization criteria such as battery State of Charge (SOC), etc. and the addition of intelligent algorithms that modify an existing control strategy based on information about the future route of the vehicle. Vehicle navigation systems these days can pinpoint the current location of the vehicle and allow planning routes to a destination. Information along that planned route may be used in the control strategy to aim at optimizing performance for an entire trip. These algorithms are currently being studied by faculty and students at the Ohio-State University.

An algorithm to produce the fuel-use maps of IC Engines is implemented in ADVISOR 3.2. This algorithm, called the Willan's Line Model, is based on the scaling of parameters from engines with similar characteristics. A database of the parameters of the Willan's Line model for SI and CI engines are included in the software. There is a provision for adding parameters of newer engines into the database if available. The advantage with this algorithm lies in the fact that the user obtains a fuel-use map, from entering some basic engine parameters such as stroke length, bore diameter, etc.

REFERENCES

1. Bernd Baumann, Gregory Washington, Brad Glenn, Giorgio Rizzoni, "Mechatronic Design and Control of Hybrid Electric Vehicles", *IEEE/ASME Transactions on Mechatronics*, Vol 5, No 1, March 2000.
2. Valerie Johnson, Kieth Wipke and David Rausen, "HEV Control Strategy for Real-Time Optimization of Fuel Economy and Emissions", *SAE 2000-01-1543*.
3. K.B. Wipke, M.R. Cuddy, S.D. Burch, "ADVISOR 2.1: A User-Friendly Advanced Power train Simulation Using a Combined Backward/Forward Approach", *NREL Technical Report*, September 1999.
4. N. Jalil, A. Kheir, M. Salman, "A rule-base energy management strategy for a series hybrid vehicle", *Proc. of American Control Conference*, pp. 689-693, June 1997.
5. Hyeoun-Dong Lee, Seung-Ki Sul, "Fuzzy Logic based Torque control strategy for parallel type HEV", *IEEE Transactions on Industrial Electronics*, vol 45, no.4, pp 625-632, 1998.
6. Bradley Charles Glenn, "Intelligent Control of Parallel Hybrid Electric Vehicles", *MS Thesis, The Ohio-State University*, 1999
7. Kevin Passino, Steve Yurkovich, "Fuzzy control," *Addison-Wesley Pub. Co.*, Sep 1997.
8. Xi Wei, Giorgio Rizzoni, "A scalable approach for energy converter modeling and supervisory control design," *Proc. of ASME IMECE'2001*, November 2001.
9. On-Star Navigation System, General Motors, www.onstar.com
10. Arun Rajagopalan, Gregory Washington, "Intelligent Control of hybrid electric Vehicles using GPS information," *SAE Future Car Congress 2002, Arlington VA*, June 2002.

APPENDIX

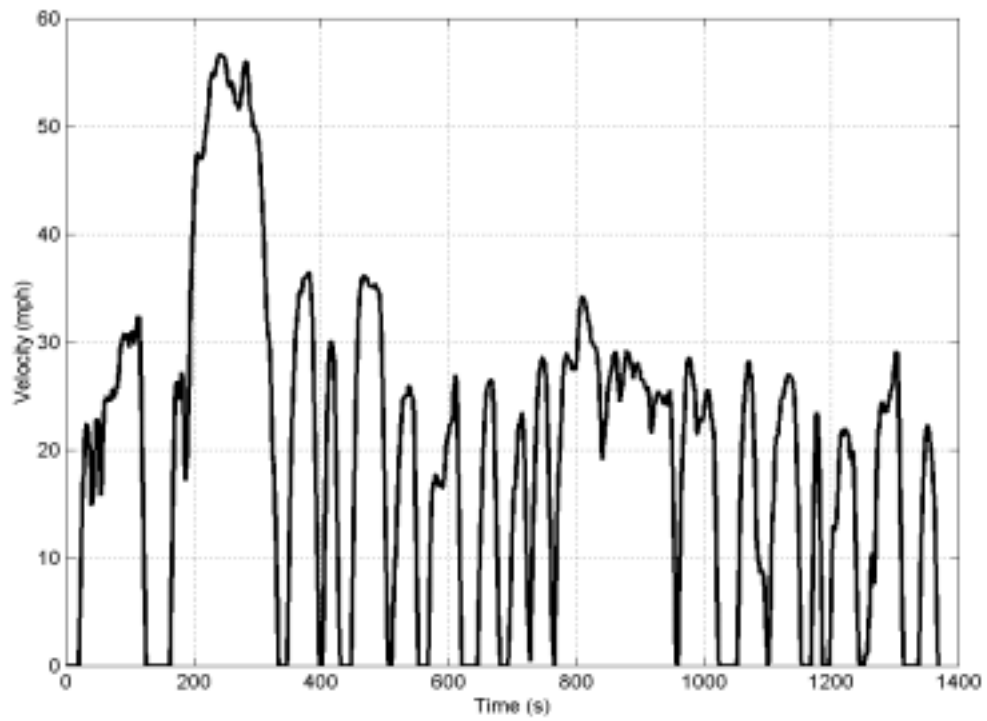


Figure 27: Urban Dynamometer Driving Schedule (UDDS)

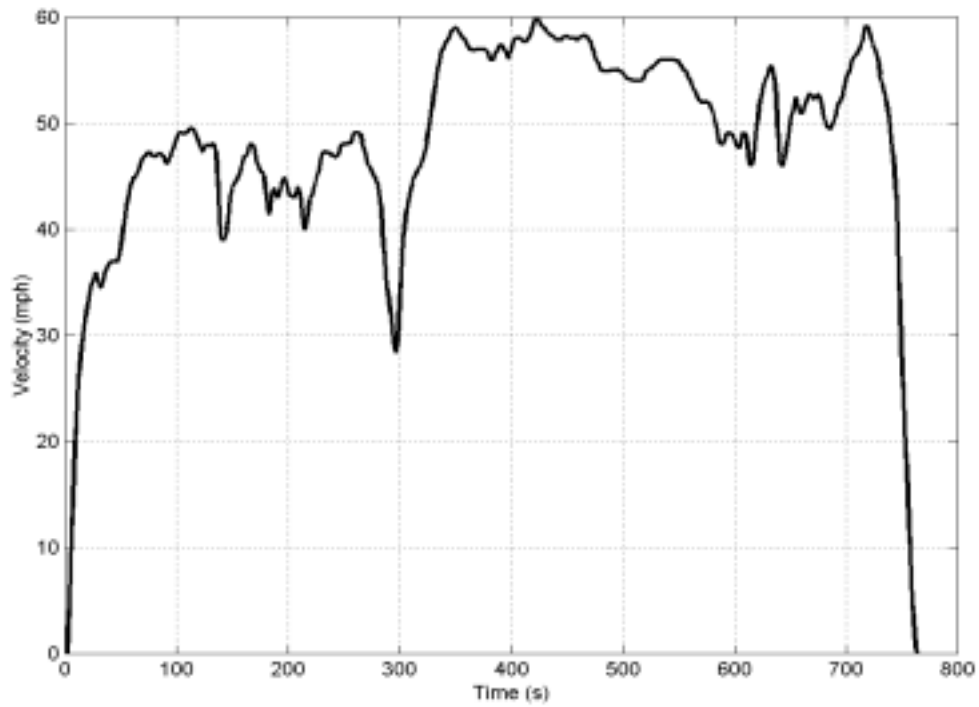


Figure 28: Highway Fuel Economy Test (HWFET)

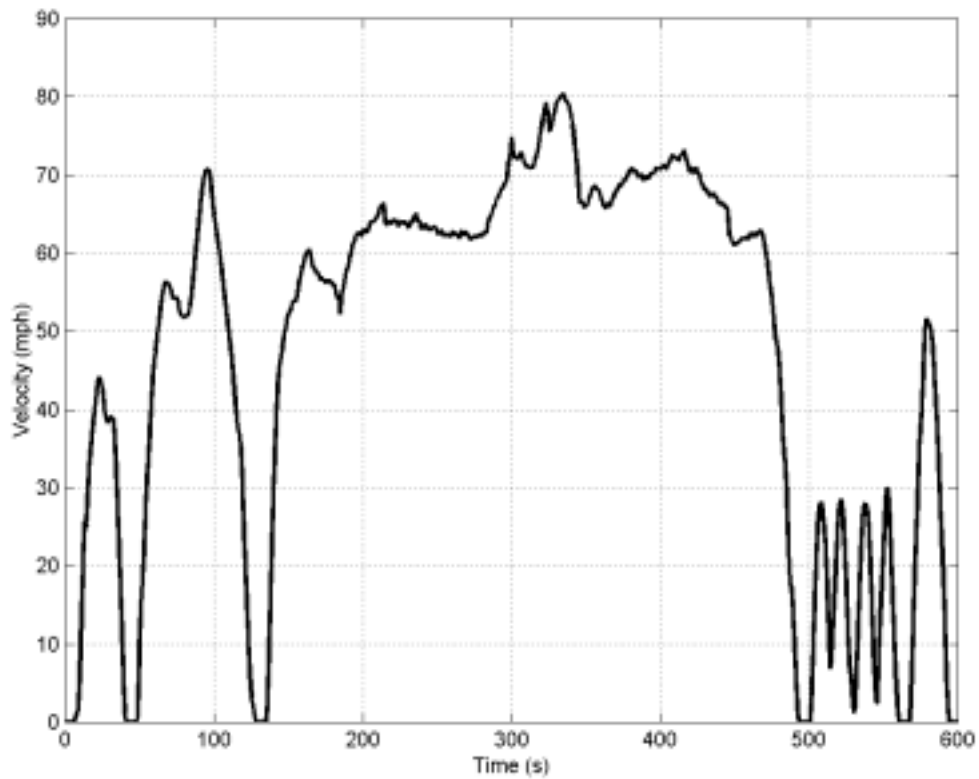


Figure 29: US06 aggressive driving cycle